SPEECH UNDERSTANDING RESEARCH

Donald E. Walker

Stanford Research Institute

Prepared for:

Advanced Research Projects Agency

February 1973

Annual Technical Report

Covering the Period 4 October 1971 through 2 October 1972

SPEECH UNDERSTANDING RESEARCH

By: DONALD E. WALKER

Prepared for:

ADVANCED RESEARCH PROJECTS AGENCY
ARLINGTON, VIRGINIA 22209

CONTRACT DAHC04-72-C-0009
ARPA Order No. 1943
Program Element Code 61101D

STANFORD RESEARCH INSTITUTE
Menlo Park, California 94025 · U.S.A.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Stanford Research Institute Menlo Park, California 94025	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP n/a

3. REPORT TITLE

SPEECH UNDERSTANDING RESEARCH

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Annual Technical Report: 4 October 1971 through 2 October 1972

5. AUTHOR(S) (First name, middle initial, last name)

Donald E. Walker

6. REPORT DATE February 1973	7a. TOTAL NO. OF PAGES ~~72~~ 61	7b. NO. OF REFS 8
8a. CONTRACT OR GRANT NO. DAHC04-72-C-0009 b. PROJECT NO. c. Program Code No. 61101D d. ARPA Order No. 1943	9a. ORIGINATOR'S REPORT NUMBER(S) SRI Project 1526 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

10. DISTRIBUTION STATEMENT

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency Arlington, Virginia 22209

13. ABSTRACT

Stanford Research Institute is participating in a major program of research on the analysis of continuous speech by computer. The goal is the development of a speech understanding system capable of engaging a human operator in a natural conversation about a specific problem domain. The approach being taken is distinctive in its use of syntactic and semantic processing to guide the acoustic analysis.

DD FORM 1 NOV 68 1473 (PAGE 1)

S/N 0101-807-6801

II-a

UNCLASSIFIED
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Natural language processing						
Question answering						
Speech understanding						
Computer understanding						
Speech recognition						
Syntax						
Semantics						

STANFORD RESEARCH INSTITUTE
Menlo Park, California 94025 · U.S.A.

Form Approved
Budget Bureau No. 22-R0293

February 1973

Annual Technical Report

Covering the Period 4 October 1971 through 2 October 1972

Stanford Research Institute Project 1526

SPEECH UNDERSTANDING RESEARCH

By

DONALD E. WALKER
Project Leader
(415) 326-6200, Ext. 3071

CONTRACT DAHC04-72-C-0009
ARPA Order No. 1943
Program Element Code 61101D

Prepared for

ADVANCED RESEARCH PROJECTS AGENCY
ARLINGTON, VIRGINIA 22209

Approved by:

BERTRAM RAPHAEL, *Director*
Artificial Intelligence Center

BONNAR COX, *Executive Director*
Information Science and Engineering Division

Copy No. 15

I

CONTENTS

ILLUSTRATIONS

TABLES

I INTRODUCTION

Stanford Research Institute is participating with other ARPA/IPT contractors in a major program of research on the analysis of continuous speech by computer* (see Newell et al., 1971†). The goal is the development of a speech understanding system capable of engaging a human operator in a natural conversation about a specific problem domain. The domain chosen by the SRI project for its initial efforts was one developed by Winograd (1971), a simulation of the actions of a robot that can manipulate various kinds of blocks. A person speaking to the computer will be able to ask questions about the "blocks world," to give commands that will modify it, and to add information to augment its structure. The procedures being developed to provide these capabilities integrate pragmatic, semantic, syntactic, lexical, phonological, phonetic, and acoustic analyses.

Efforts toward speech understanding contrast with those directed toward speech recognition both in goal and in approach. Speech recognition work has aimed at providing an orthographic transcription of the sounds and words corresponding to the speech signal. Analysis has concentrated on acoustic processing, although linguistic segmentations have been attempted, particularly in relation to phonetics, phonemics, and morphology. In contrast, speech understanding research seeks to determine the message intended in relation to the accomplishment of some task, in spite of indeterminacies and errors in the generation, transmission,

*Contract No. DAHC04-72-C-0009, SRI Project 1526.

†References are listed alphabetically at the end of the report.

and reception of an utterance. Special emphasis is placed on semantic, syntactic, and pragmatic information, and a question-answering system may be used as a major processing component. In particular, our approach at SRI stresses the critical role of semantics and pragmatics in reducing the amount of acoustic processing necessary to understand an utterance.

The ARPA Speech Understanding Research Program spans the broad range of research and development efforts necessary to produce a prototype speech understanding system. Within this range, our activities at SRI are directed toward the following goals:

- Implementation of an integrated system
- Establishment of effective procedures for handling semantic and pragmatic information
- Formulation of techniques for dealing with variability within and among speakers.

Our strategy in pursuing these goals has been to put a system into operation at the earliest possible time, making maximum use--within the confines of the basic system design concept--of existing programs and algorithms. We believe that exercising a system will allow us to identify the kinds of links that need to be established among the components, as well as provide useful guidance for constructive revision of the components themselves.

In the system we are developing at SRI, knowledge of the world, a model of the user, and a grammar will combine to constrain the selection of a small set of words, each of which might be expected to be present at a particular place in the speech stream representing an utterance. For each word, a program is written to enable determination of how well the word corresponds to the acoustic data for that place. When the presence of a word is confirmed, a new set of words is selected for testing at the next place in the speech stream. Successive steps through the utterance result in a determination of its structure.

During this first year of the project, development has been carried to the point where syntactic, semantic, and acoustic data are used in processing sentences. The capabilities developed are rudimentary, but we can predict words and test for their presence. No model of the user has been developed yet. More preprocessing of the acoustic data is done than we believe will be necessary. Only a small number of word functions have been written; thus, it is not possible to step through a complete utterance. Nevertheless, the results to date are sufficient to encourage us to continue implementation of the system design.

A description of the current state of the system is presented in Section II. A detailed analysis of the system components is given in Section III. Section IV considers the directions for continuing research and development.

II AN INTEGRATED SYSTEM FOR SPEECH UNDERSTANDING

A. Introduction

In the current state of the SRI system for speech understanding, it is possible to identify three major components: Pintle, a set of procedures for syntactic and semantic analysis; programs for acoustic processing; and a word verifier routine that links the other two. There will be additional components and major changes in all three of the present ones, as well as much more complex interrelationships. Nevertheless, in its current state, the system does illustrate an approach to speech understanding that is distinctive because of its dependence on syntactic and semantic processing.

Pintle is a major modification (in ways described below) of Terry Winograd's system for procedural analysis of language (Winograd, 1971). It combines a grammar--written as a set of programs--with semantic routines that model changes in the arrangement of a set of blocks. A sentence constitutes a path through the grammar. Branching at choice points is determined by the order of the rules, by features on other constituents, and by semantic data. At the end of each branch in the parse tree is a set of words from a particular grammatical class (e.g., determiners, adjectives, nouns, verbs), from which a subset can be selected on semantic grounds.

The acoustic routines convert the recorded analog voice input to digital form. The digitized signal is then fed into a bank of digital filters, which make it possible to assign successive acoustic segments to the following rough classes: silence, voiced turbulence, unvoiced turbulence, voiced stop, vowel-like, or other. The signal also is

processed by a more complex acoustic analysis procedure that identifies the frequency and amplitude for the first three spectral peaks of the vowel-like sounds. The data from these two analyses are stored in files.

The word verification routines take a set of words produced by Pintle and test each word against the acoustic data for a particular portion of the utterance. The result is a subset of the words, ordered according to agreement with the acoustic data, with each word containing a pointer to identify its approximate endpoint in the acoustic stream. Pintle takes the most likely word first and then proceeds on its path through the grammar to select the next set of words for processing by the word verifier. Testing this new set against the acoustic data begins at the point designated by the endpoint for the word previously accepted.

An example is considered next, and a more detailed description of each of the components is presented in Section III.

B. Understanding a Sample Sentence

A brief description of the "blocks world" problem domain used in the SRI system is necessary as background for the analysis of the sample sentence. Visualize a table containing a box and several objects of different sizes, shapes, and colors. There are five blocks (two red, one black, one green, one blue) and three pyramids (green, blue, and red); the box is white. The objects are arranged in a particular configuration in the computer representation of the scene, but the details of the arrangement are not necessary to understand the example. Commands given to a simulated robot arm cause it to move the blocks. Alternatively, the person interacting with the system can ask questions or provide information that will augment or change the semantic structure of the world in some way.

The sentence to be processed is the following:

PUT THE BLACK BLOCK IN THE BOX.

All of the steps involved in its analysis are presented in the following pages as they occurred in an actual demonstration. The capabilities shown reflect the state of the system as of September 1972. Lines prefixed by an arrow represent entries by the user. Figure 1 contains the actual protocol.

BBN LISP-10 12-11-72

The speech understanding system is implemented in BBN-LISP and is run on a PDP-10 computer under the Tenex Operating System.

←SYSIN(⟨ROBINSON⟩SPINTLE)

The file containing the system is called to be read in.

(⟨ROBINSON⟩SPINTLE.;1)

Confirmation that the system (Version 1) is read in.

←DEMOTRACE)

The trace is turned on to show the output sequentially.

AINTERP

Confirmation.

←FKINIT(⟨RIDDER⟩LSWDS.SAV)

The FORTRAN fork containing the acoustic routines is initialized. (The LISP-FORTRAN interface allowing access to FORTRAN data files from LISP will be described later.)

⟨RIDDER⟩LSWDS.SAV

Confirmation.

←SPEECHDATA(M351R)

The prerecorded acoustic data for the sample sentence are read in.

NIL

Confirmation.

←# (PUT THE ($ 37) ($ *) IN THE ($ 134).)

Analysis of the sentence by Pintle is initiated. At the time this

```
BBN LISP-10  12-11-72 ...

←SYSIN(<ROBINSON>SPINTLE)
(<ROBINSON>SPINTLE.;1)
←DEMOTRACE)
AINTERP
←FKINIT(<RIDDER>LSWDS.SAV)
<RIDDER>LSWDS.SAV
←SPEECHDATA(M351R)
NIL
←# (PUT THE ($ 37) ($ *) IN THE ($ 134).)
...
PUT
THE
BLACK
BLOCK
IN
THE
BLACK
BOX

(CLAUSE MAJOR IMPER ACTV TRANSL)
     (VG IMPER)
          PUT (INF PAST VB TRANSL VPRT MVB)
     (NG OBJ OBJ1 NOLOC DET DEF NS)
          THE (DET NPL NS DEF)
          BLACK (ADJ)
          BLOCK (NOUN NS)
     (PREPG PLACE LOBJ)
          IN (PLACE PREP)
          (NG OBJ PREPOBJ DET DEF NS)
               THE (DET NPL NS DEF)
               BOX (NOUN NS)

*
MOVETO 472 192 128
GRASP #B3
MOVETO 448 448 129
UNGRASP
  OK.

NIL
←
```

FIGURE 1 ANALYSIS OF A SAMPLE SENTENCE

protocol was made, the word verifier did not have word functions available for the sets of words including PUT, THE or IN. Under these circumstances--and in general to allow more flexible testing of the system--it is possible to enter text to specify a word. For convenience, the word verifier checks the input text first to see whether any of the words in the set predicted has been typed in. Finding none, it will use the appropriate word functions, if they are available. If none are present, the words in the set will be rejected.

PUT

Pintle begins by looking for a major clause; branching along the imperative path, it looks for command verbs. The word PUT is among those in the set generated at this point, and it is found in the text input.

THE

Having found a verb, Pintle begins its search for a noun group by looking for a determiner. THE is confirmed from the text input.

BLACK

Having found a determiner, Pintle looks for an adjective. Since there is no text input, the predicted words are tested against the acoustic data. Beginning at a location 370 milliseconds into the utterance (the 37th 10-millisecond segment), the word verifier finds that BLACK corresponds to the acoustic data at the highest of four confidence levels; RED, GREEN, BLUE, and WHITE are rejected.

BLOCK

Pintle now looks for a noun to complete the noun group. It begins at the location in the acoustic data confirmed as the ending place for BLACK, the previous word accepted. This condition is specified by the asterisk in the initial entry for the sentence to be processed. The word verifier finds that BLOCK corresponds to the acoustic data

at the highest confidence level; BALL, PYRAMID, and THING are rejected.

IN

With the noun group complete, Pintle looks for a prepositional phrase to complete the action denoted by PUT. IN is confirmed from the text input.

THE

Looking for a noun group to complete the prepositional phrase, Pintle begins with the determiners. THE is confirmed from the text input.

BLACK

Pintle next looks for adjectives. Beginning at a location 1340 milliseconds into the utterance (the 134th segment), the word verifier finds that BLACK corresponds to the acoustic data at the third highest confidence level. RED, GREEN, BLUE, and WHITE are rejected.

BOX

The location returned along with BLACK proves to be the end of the utterance. Consequently, BLACK is rejected because the string so ending would be ungrammatical; it also would be semantically unacceptable. Pintle backtracks and begins looking for nouns. BOX corresponds to the acoustic data at the highest confidence level; BALL, PYRAMID, and THING are rejected.

(PUT THE BLACK BLOCK IN THE BOX)

The sentence as identified. At this point the system returns a printout showing the grammatical structure of the sentence.

```
(CLAUSE MAJOR IMPER ACTV TRANSL)
     (VG IMPER)
          PUT (INF PAST VB TRANS VPRT MVB)
     (NG OBJ OBJ1 NOLOC DET DEF NS)
          THE (DET NPL NS DEF)
          BLACK (ADJ)
```

BLOCK (NOUN NS)

(PREPG PLACE LOBJ)

IN (PLACE PREP)

(NG OBJ PREPOBJ DET DEF NS)

THE (DET NPL NS DEF)

BOX (NOUN NS)

Briefly summarized, the sentence is identified as an active imperative with the verb PUT involving the movement of an object, THE BLACK BLOCK, to a location, IN THE BOX. (See the next section and Winograd, 1971, for more extended discussions of the grammatical structure.) This information is used to direct movements of the simulated robot arm in conjunction with the semantic definitions for the words and the determination of their relevance for the current state of the problem domain, as indicated in the following steps.

MOVETO 472 192 128

The arm moves to the black block, specified by the three coordinates given.

GRASP :B3

The arm grasps B3, the black block.

MOVETO 448 448 129

The arm with the black block moves to the box, specified by the three coordinates.

UNGRASP

The arm releases the black block inside the box.

OK

The system indicates that it has completed the action identified in its "understanding" of the input utterance.

C. Provisional Features of the Current Implementation

In anticipation of a more detailed discussion of the system components in the next section, it is probably reasonable to note here some characteristics of the analysis of the sample sentence that are temporary expediencies not to be considered characteristic of the system design. Some of the remarks made here are amplified in Section IV, Directions for Continuing Research and Development.

The system is not totally on-line at this time; i.e., it is not possible to speak directly into the system and to initiate processing accordingly. Analog-to-digital conversion of the speech signal cannot be performed on our PDP-10/15 computer facility yet, pending completion of the necessary software. Consequently, for the present we are digitizing the signal on a PDP-11 and transferring the resulting files by tape to the PDP-10 for the remainder of the acoustic analysis.

The FORTRAN files accessed by the word verification routines now contain preprocessed data from both the digital filters and from the linear predictive coding analysis. In the system as designed, we expect to do an analysis in real time that will produce the preliminary classification of acoustic segments now provided by the digital filters. However, we expect to perform spectral analyses of the kind provided by LPCs and to call for other complicated acoustic processing only as required to make the kinds of decisions necessary to distinguish among the predicted words in relation to the acoustic data.

As noted in the analysis of the sample sentence, only a small number of word functions have been written. Consequently, it is not yet possible to process a complete sentence. The option of testing predicted words against textual, as well as acoustic, data is useful for debugging the acoustic routines for particular sets of words. It is also useful in the absence of semantic and prosodic procedures for establishing

constraints on paths through the grammar at the beginning of utterances, and, in particular, at the beginning of a dialog when no context has been established.

A final comment on the analysis embodied in the sample sentence is probably in order. We have not exercised the system to any great extent. There are only a few word functions, and they have been tested against only two speakers. The flow of control in the current implementation is primarily from the syntactic and semantic component to the acoustic. It is clear, however, that useful information can pass in the opposite direction, not only from what a prosodic analysis might provide, but also from what might be expected to arise in the course of testing words against the acoustic data. In addition, there probably are ways in which the word verifier, which currently processes one word at a time, can operate more efficiently on the whole set of predicted words in relation to the acoustic data, thus reducing the search space involved.

III SPEECH UNDERSTANDING SYSTEM COMPONENTS

A. Initial Resources

When the project was initiated at SRI, we had available (or could quickly obtain):

- Techniques for analog-to-digital conversion of the speech signal.
- Algorithms for performing Fast Fourier Transforms (FFTs) to provide spectral data.
- The beginnings of an interactive man/machine speech analysis facility for use as a research and development tool.
- The bases for two approaches to syntactic and semantic processing.
  - A question-answering system, based on the first-order predicate calculus and incorporating a resolution theorem prover, which has been evolving at SRI over a period of years.
  - A program package, originally developed at MIT, for the Winograd system for natural language understanding.

Our plan for the first year included the following tasks:

- To develop a comprehensive capability for acoustic analysis, including completion of the interactive speech analysis system.
- To pursue the two approaches to syntactic and semantic analysis, modifying each to allow for speech input.
- To prepare for each syntactic and semantic component an interface with algorithms built on the acoustic analysis processes.
- To implement these interfaces and test the resulting systems.
- To do the necessary work on system organization required to coordinate LISP and FORTRAN programs and provide a common file structure that would allow data to be accessed by each, while exploring the relevance of QA4, a new programming system being developed at SRI under separate support (Rulifson, 1972),* for future system implementations.

*Contract NASW-2086, SRI Project 8721

We were able to carry out this plan during the year with one exception. It was not possible to prepare algorithms that would allow the SRI question-answering system to interface with acoustic processes. Not only would that effort have required more resources than were available, but, in addition, similar algorithms are being developed elsewhere, and it should be possible to take advantage of the developments later. The decision to concentrate on the system described in Section II was made because it involved a more radical design concept and because the requirements for acoutsic analysis are believed to be less demanding. The present section contains more detailed descriptions of the components of this system. Also included are a brief overview of the interactive speech analysis facility and a summary of the work performed on the SRI question-answering system in anticipation of an acoustic interface.

B. Pintle--Procedures for Syntactic and Semantic Analysis

Pintle, the syntactic and semantic component of the SRI system for speech understanding, is based currently on the Winograd "Computer Program for Understanding Natural Language" (Winograd, 1971). It is a top-down system for linguistic analysis in which syntax, semantics, and inference are combined to direct the processing of questions, statements, and commands. Now implemented by SRI in BBN-LISP, Pintle constitutes a substantial modification of Winograd's program. Changes have been made in the linguistic analysis, in the ordering of paths in the grammar, in the flow of control, and in the establishment of semantic constraints. A backtracking facility has also been introduced.

In Winograd's work, as in most existing parsing systems, successive words from a typed input string guide the analysis. Since we proposed to use the parsing procedure to help segment and identify the words in the speech input, it was necessary to find other ways to control the generation of paths through the grammar. In order to explain the

operation of Pintle, it will be useful to consider the general form of that grammar.

Michael Halliday's systemic grammar forms the linguistic basis for Winograd's system, and we have used it in our initial work with only minor modifications (see Winograd, 1971, and Hudson, 1971, for detailed descriptions). In Halliday's grammar, syntactic and semantic features are associated with words and with higher order grammatical structures. There are three basic ranks of units: word, group, and clause. The word is the basic constituent; the word classes include noun, verb, adjective, determiner, preposition, among others. There are four groups: noun group, verb group, preposition group, and adjective group, each of which has slots for the words that compose it. For example, one noun group might include determiner, number, adjective, classifier, and noun. A clause can be major or secondary; major clauses may be declarative, imperative, or question, active or passive, and the like; secondary clauses account for relatives, complements and various kinds of modifiers and qualifiers. A unit at any rank has associated with it a set of features. For example, words exhibit features identifying number, inflection, various kinds of affixation; groups may show definiteness, tense, negation; clauses may be marked for yes-no or WH questions, subject or object. There are systems of mutually exclusive features and networks representing the relations among the units at a rank.

Each of the units above the word level (clause, noun group, verb group, preposition group, adjective group) is represented in Winograd's system by a program written in PROGRAMMAR, a language developed by him for this purpose. Parsing is done by an interpreter that processes PROGRAMMAR code; the flexibility of this method allows various kinds of tests to be made that call on larger grammatical contexts and on other sources of information, particularly semantic ones. The parser operates

in a top-down, left-to-right manner, beginning with a search for a major clause. In Winograd's implementation, the clause program checks the features of the first word in the typed input string to decide what unit to begin with. Features in words guide the parser through an analysis by delimiting or selecting subsets of related groups of choices. In this way, the parser traces a path through the grammar, arriving at a structure for the sentence.

To adapt Winograd's procedures for speech understanding, it was necessary to establish syntactic and semantic constraints that influence successive choices through the grammar, leading to the selection of a subset of the words of a particular word class. In what follows we are presenting the information available in the grammar for this purpose, with some additions where it seemed appropriate. Consider again the sample sentence discussed in the previous section, PUT THE BLACK BLOCK IN THE BOX. Assuming that at the time this utterance is made in a hypothetical dialog of a user with the system it is reasonable to expect a command, the clause program would look for an imperative. (Prosodic information also may provide such guidance.) Since imperative clauses generally start with verbs, the parser enters a verb group program looking for imperatives. Since imperatives are in infinitive form, only those verbs with that feature are identified. The result of this path through the grammar is a small set of imperative verbs, one of which may correspond to the first word of the utterance. We expect to be able to constrain the set of verbs further by additional semantic information--perhaps regarding what command might be appropriate at this point in the dialog. And pragmatic information specific to a particular user should be possible to capture; for example, the frequent use of certain commands. However this verb group is constrained, the initial result is a set of words to check against the acoustic data.

Confirming one (or more) of the words from this initial set might result in Pintle looking for a noun group, as is the case with the word PUT, which requires an object. Identification of a different imperative, PICK, could result in Pintle looking first for the particle UP. Accepting PUT in the sample sentence, Pintle might begin the search for a noun group with a determiner. Since the set of determiners is small, all of them could be predicted. However, they are difficult to distinguish acoustically, and it might be reasonable, on semantic grounds, to look only for a definite or only for an indefinite determiner, e.g., THE or A.

Finding a determiner, an adjective would be likely to follow. There are various classes of adjectives, and in English there is an ordering controlling the sequence in which they typically modify a noun. For instance, size adjectives precede color adjectives; e.g., BIG RED BLOCK but not RED BIG BLOCK. Again, in a dialog it would be reasonable at certain points to predict the amount of specificity required to identify an object on the basis of its qualities. Pragmatically, (assuming we have models for our users) some people may make things perfectly clear, while others are more sparing in their characterizations. So, sets of adjectives will be checked against the acoustic data. Subsequently, and in a similar fashion, various paths among the nouns would be selected for testing. The kind of verb would influence the choice; verbs of manipulation call for nouns that represent manipulable objects. This information also could be used to influence the choice of an adjective in the prior search, limiting it to those adjectives appropriate to manipulate objects.

Continuing the parse beyond the noun group would lead to consideration of preposition groups because PUT requires a location. Identifying a place preposition would lead to a search for an object noun group, with decisions being made similar to those discussed for the preceding noun group. However, only those nouns that can have objects PUT IN them need

to be considered. In this manner, a set of predictions are made regarding the sequence of sets of words likely to occur in the utterance.

The foregoing description presumes the accuracy of the initial predictions. In the sample sentence, however, the adjective initially found in the second noun group proved to be in error. Thus, backtracking and tracing down an alternate path were required to find the noun. An interpreter for PROGRAMMAR has been added that contains a backtracking mechanism not available in Winograd's system. The interpreter makes it possible to specify a set of alternatives at a particular point in the grammar and to try these in succession, backtracking automatically if the initial choice is not subsequently confirmed. This same mechanism allows an easy return and recovery following acceptance of a word that proves to be in error, as in the sample sentence.

The requirement for speech input (the absence of words with identifiable features in the input string) and the availability of the backtracking facility resulted in other modifications to Winograd's analysis procedure. Winograd tested to eliminate the least likely alternatives first, checking the longest possible constituent and cutting back wher that failed. PROGRAMMAR, in his original version, returned the first successful analysis, having provided both syntactic and semantic guidance to make that a likely interpretation within the model of the "blocks world." Selective backup was possible in a particular situation, but it involved specifying a location to return to for alternative processing. With voice input, it is necessary both to test for most likely alternatives first and to have a more general backup mechanism in case of failure. What is needed further for speech understanding is the flexibility in the grammar to allow dynamic reordering of rules, depending on the state of the analysis at the moment. To help provide this capability, changes have been made that allow identification at any particular choice point in the grammar of what alternatives are possible.

In Winograd's system, alternative choices could only be identified serially after failure of the predecessor.

Many more changes in Pintle are contemplated that will improve its ability to use syntactic, semantic, and--hopefully--pragmatic constraints to reduce the number of words that needs to be considered at any particular point. Currently, checking against the actual configuration of objects on the "blocks world" is done only after a group has been parsed. Thus, in the sample sentence, both BALL and PYRAMID are tested against the acoustic data. However, there are no balls in the current situation (although the word is in the lexicon), and there are no black pyramids. Information of this kind can and should be used to influence the selection of words in a set as soon as it is relevant.

Major modifications in Pintle can be expected to follow the introduction of new structures for managing semantic and pragmatic information. These new structures will replace and extend Winograd's MICROPLANNER code for "blocks world" manipulation, Exploratory development will be done in QA4, a procedure-oriented programming system particularly well suited for work in artificial intelligence because of its flexibility and special features. We also plan to experiment with revisions to the parser in QA4; new techniques are necessary to facilitate the accommodation of semantic and pragmatic information and to simplify the dynamic reordering of paths through the grammar.

C. A Grammar for the Speech Understanding System

No substantial changes have been made as yet in the actual grammatical rules that Winograd has in his system. However, modifications are essential because there are significant differences between spoken and written English. Previous systems for computational linguistic analysis, including Winograd's, have worked with grammar rules for the written

language. But, function words and affixes, which existing parsers use extensively for structural identification, tend to be blurred in speech. Instead, prosodic features--such as intonation, stress, pause, and juncture--are used as indicators and delimiters. Spoken utterances are frequently incomplete and include errors, hesitations, and false starts, all of which are either edited out of the typed input or relatively easy to identify in it. In contrast, there is no easy way to separate out the well-formed parts in speech. Misspelled words are easy to detect; mispronounced words are not. Moreover, the relation between a word and its alphabetic representation is stable in text, whereas the spoken form of a word varies significantly and often dramatically in relation to other words around it. Phonetic or even phonemic transcriptions are not sufficient. Consequently, it is essential to have a set of grammatical rules suitable for working with spoken English.

A group at the University of Michigan under the direction of Michael O'Malley has begun work on a grammar of spoken English, and we are working closely with them in its development. Our intent is to incorporate prosodic information directly into our procedures to help determine sentence type, to identify phrase and clause boundaries, to eliminate false paths, to reduce ambiguity, and to provide a basis for handling incomplete sentences and hesitations. More recently, groups at UNIVAC and at the Speech Communication Research Laboratory in Santa Barbara have joined in the study of prosodic information.

In conjunction with the University of Michigan and with Bolt Beranek and Newman, (BBN) we have begun a comparative analysis and evaluation of the current grammars used by BBN and SRI. We hope to clarify similarities and differences and to establish a common grammar for the overall ARPA Speech Understanding Research Program. As the first step in this effort, we converted the grammar in Pintle into the transition network formalism used by BBN to facilitate comparison.

Since this model seems to be easier for linguists to work with, having this alternative representation for our grammar may simplify incorporating the prosodic rules.

A complete grammar for spoken English should include a phonological component that contains acoustic/phonetic rules relevant for the generation and recognition of utterances. It is not clear that all of the strategies for speech understanding could make productive use of such a grammar, but it could serve to model a substantial part of the relevant linguistic framework.

In addition to developing a set of rules for a grammar of spoken English sentences, a considerable amount of research needs to be done to provide structural descriptions for larger amounts of discourse, particularly those involved in dialog of the kind envisioned for interaction with the various problem domains selected by the ARPA contractors.

D. Acoustic Processing

The major capabilities for acoustic processing added to our facility for use in speech understanding were a digital filter package and a procedure for calculating linear predictive coefficients (LPCs). The decision to use digital filters rather than to build a hardware filter bank was made because of our uncertainty about the most appropriate set of filters for the system. The ones currently implemented were chosen to allow us to make a preliminary classification of segments, as described below. Changes can be made easily to refine that classification, and it will be simple to specify an analog filter package or parallel digital filters when we need to be concerned about time required for processing. The particular filter package we acquired is described in more detail in Appendix A.

The programs we have written for linear predictive coding, essential-

ly John Markel's algorithm (Markel, 1971), produce better spectral data than could be gotten from Fourier analysis (FFTs) and can provide a major portion of the frequency analysis done by conventional analog or digital filters. We have been using LPCs successfully for formant tracking, and we are developing a pitch-extraction procedure based on them.

An overview of the acoustic data processing currently done in the SRI system is presented in Figure 2. The speech data are obtained in a quiet room using a B&K 433 condenser microphone and an Ampex AG 500 tape recorder. An analog tape is produced at 7-1/2 inches per second recording speed. The speech data on the tape are then digitized in segments of up to 3.1 seconds in length. A presampling low-pass filter with an 8-kHz bandwidth is employed to reduce aliasing errors, and the digitization is accomplished by 12-bit A/D converter operating at a rate of 20,000 samples per second.

The raw digital data are processed further by digital filtering and by linear predictive coefficient analysis. Figure 2 indicates that five rms values of the time series data are calculated in each 10-millisecond interval of time. Four of these values are from time series calculated by digital filters with bandpass characteristics shown on the figure. The fifth value is calculated from the unfiltered time series. Each of these five values is labeled by the notation shown in parentheses, e.g., raw, voice. The upper channel indicated on Figure 2 calculates three formant frequencies and amplitudes by finding spectral peaks in a 128-point spectrum derived by an LPC analysis of the raw time series data. The spectral peak data, which correspond to formants in voiced speech segments, are stored immediately on magnetic disk files for later use.

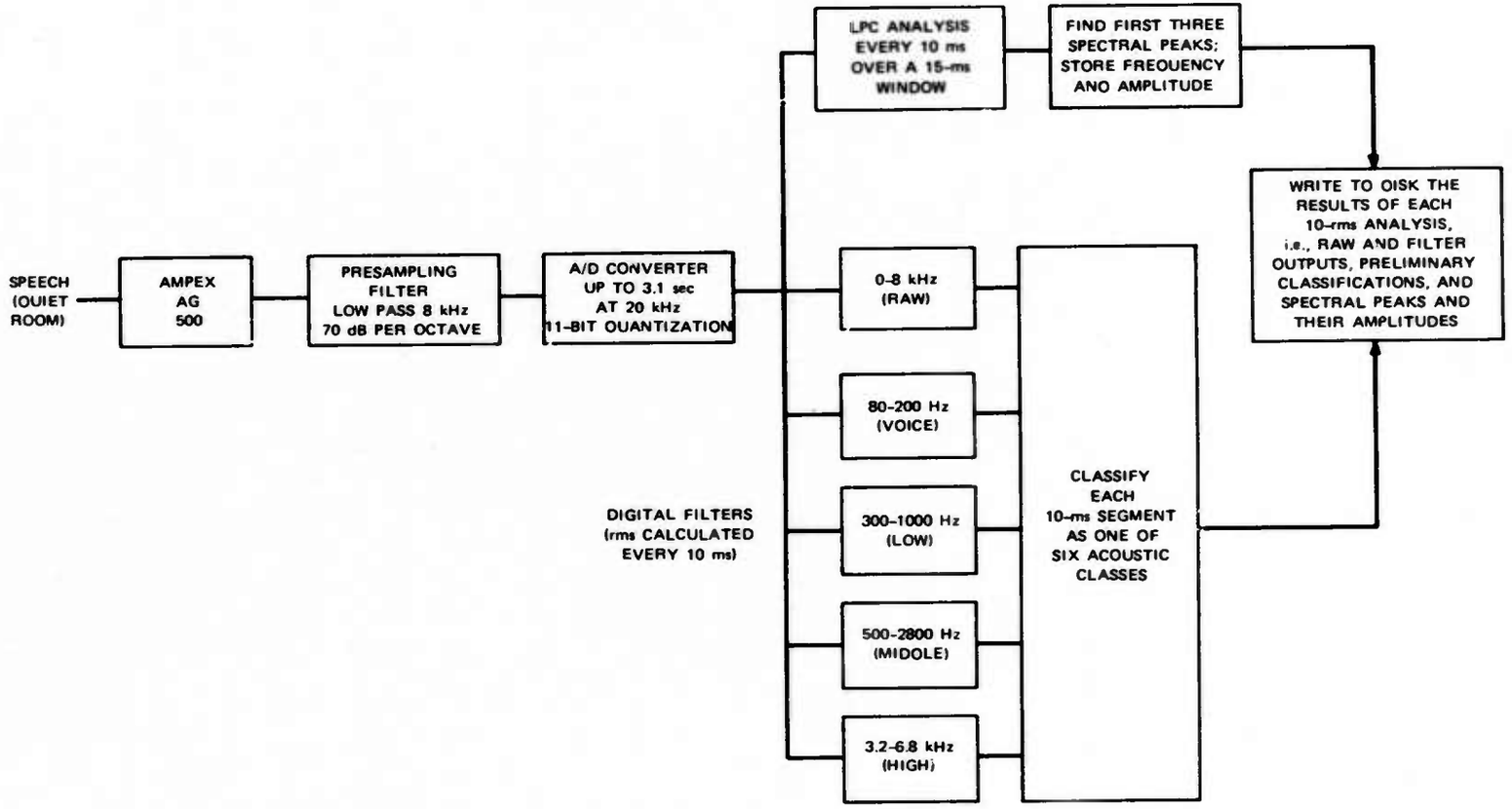

FIGURE 2 BASIC ACOUSTIC PROCESSING FOR THE SRI SPEECH UNDERSTANDING SYSTEM

The strings of rms values are used in a classification algorithm shown in Figure 3. This algorithm currently classifies each 10-millisecond time segment as one of six events:

- Silence, SI
- Unvoiced turbulence, UT
- Voiced turbulence, VT
- Voiced stop, VS
- Vowel-like, V
- None of the above, T.

The filter outputs and preliminary classifications of each segment are stored in disk files with the formant frequency and amplitude data, and they are subsequently available for further processing.

Table 1 summarizes the extent of the current acoustic processing on the utterance PUT THE BLACK BLOCK IN THE BOX. Reading any line from left to right, the entries have the following meanings:

- Column 1 is the time in milliseconds of the occurrence of the segment being analyzed and described by the line.
- Column 2 is the segment classification, one of six class names.
- Columns 3 through 7 are pairs of rms and decibel values of filter outputs.
- Column 8 is identical to Column 2.
- Columns 9, 10, and 11 are the frequencies of the first three spectral peaks.
- Columns 13, 14, and 15 are spectral peak amplitudes corresponding to the frequencies listed in Columns 9, 10, and 11.
- Column 12 is an overall rms value in decibels computed from the LPC analysis. The numbers differ slightly from the raw value in Column 3 because there is a difference in the interval of time over which the rms is computed.

Note that Columns 3,4,5,6, and 7 each contain two values. The first is an rms value computed on a 10-millisecond time series, and the

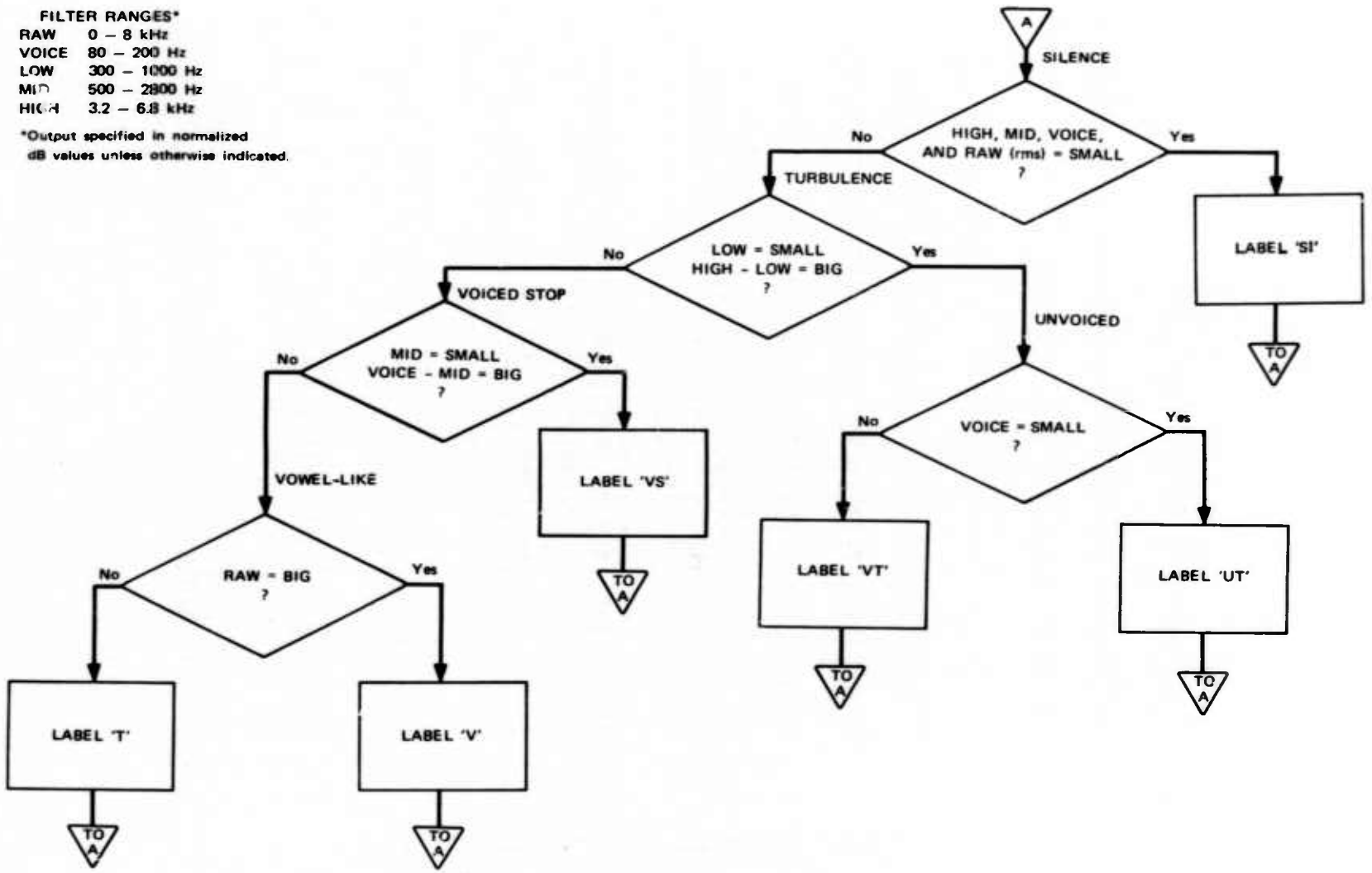

FIGURE 3 CLASSIFICATION ALGORITHM FOR ACOUSTIC SEGMENTS

Table 1

ACOUSTIC DATA FOR THE SAMPLE SENTENCE

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
TIME	CL	RAW	VOICE	LOW	MID	HIGH	CL	F1	F2	F3	RMS	A1	A2	A3
10	SI	1.-56.0	0.-44.7	0.-50.0	0.-50.0	0.-41.0	SI	195.	1326.	2145.	0.0	-10.0	-4.0	-4.0
20	SI	1.-50.0	0.-44.3	0.-50.0	0.-50.0	0.-41.9	SI	1053.	2379.	3783.	0.0	-4.0	-1.0	5.0
30	SI	1.-46.3	1.-41.9	0.-50.0	0.-50.0	0.-41.1	SI	1170.	1911.	2769.	0.0	-7.0	-2.0	-4.0
40	SI	1.-48.6	1.-42.3	0.-50.0	0.-50.0	0.-40.8	SI	1289.	1911.	2769.	0.0	-7.0	-11.0	13.0
50	SI	1.-50.0	1.-41.4	0.-50.0	0.-50.0	0.-40.4	SI	195.	1521.	2691.	0.0	-15.0	-3.0	-6.0
60	SI	1.-47.9	1.-39.0	0.-50.0	0.-50.0	0.-40.2	SI	1053.	1716.	3003.	0.0	1.0	-5.0	-1.0
70	SI	1.-50.0	1.-41.3	0.-50.0	0.-50.0	0.-41.5	SI	1053.	1794.	2652.	0.0	0.0	-5.0	0.0
80	SI	1.-50.0	1.-41.3	0.-50.0	0.-50.0	0.-41.6	SI	1053.	1833.	2340.	0.0	-9.0	-3.0	-9.0
90	SI	1.-50.0	1.-42.3	0.-50.0	0.-50.0	0.-39.2	SI	1599.	2457.	3159.	0.0	-12.0	0.0	-4.0
100	SI	1.-50.0	1.-42.4	0.-50.0	0.-50.0	0.-41.4	SI	819.	1560.	2340.	0.0	-10.0	14.0	-3.0
110	SI	1.-50.0	1.-40.4	0.-50.0	0.-50.0	0.-41.4	SI	1209.	2691.	3510.	0.0	-7.0	-6.0	0.0
120	SI	3.-48.2	0.-44.7	1.-45.1	2.-45.9	1.-32.0	SI	507.	1911.	2535.	0.0	7.0	0.0	5.0
130	T	6.-35.2	2.-32.7	4.-34.0	5.-36.4	1.-30.8	T	624.	1248.	1755.	0.0	1.0	14.0	6.0
140	T	5.-36.6	2.-33.3	2.-48.4	4.-37.7	0.-37.5	T	1170.	1482.	2223.	0.0	15.0	0.0	-4.0
150	T	4.-39.6	1.-35.2	2.-41.3	3.-41.4	1.-34.7	T	585.	1248.	2535.	0.0	-6.0	12.0	44.0
160	T	4.-39.3	2.-33.1	2.-43.8	2.-42.8	1.-36.8	T	1365.	2496.	3159.	0.0	19.0	-5.0	-4.0
170	V	67.-14.4	18.-11.8	43.-14.8	31.-20.6	2.-25.7	V	468.	1326.	2067.	0.0	22.0	21.0	0.0
180	V	149. -7.7	65. -8.6	105. -7.0	73.-13.1	8.-12.7	V	468.	1404.	2145.	0.0	16.0	17.0	1.0
190	V	162. -6.8	73. -8.8	132. -5.8	86.-11.7	14. -7.9	V	468.	1482.	2613.	0.0	0.0	20.0	15.0
200	V	169. -6.4	67. -8.4	127. -5.3	83.-11.9	16. -7.8	V	468.	1482.	2574.	0.0	0.0	14.0	10.0
210	V	112. -9.9	67. -8.4	121. -7.4	99.-15.5	13. -8.6	V	351.	1443.	2535.	0.0	16.0	9.0	6.0
220	V	38.-19.3	44. -4.1	15.-24.1	6.-34.4	2.-24.7	V	351.	1404.	2496.	0.0	15.0	-6.0	0.0
230	VS	28.-21.9	28. -7.8	10.-27.5	3.-41.9	0.-37.3	VS	312.	1443.	2457.	0.0	14.0	-3.0	8.0
240	VS	59.-15.6	22. -9.9	2.-39.5	1.-50.0	0.-40.6	VS	390.	1716.	3042.	0.0	0.0	-7.0	-2.0
250	VS	62.-15.3	14.-14.2	3.-38.3	1.-49.9	1.-35.8	VS	390.	1560.	2613.	0.0	19.0	-1.0	-4.0
260	VS	33.-28.5	5.-22.7	1.-44.1	1.-50.0	0.-38.8	VS	117.	1443.	2340.	0.0	9.0	4.0	-7.0
270	VS	9.-31.5	11.-16.1	1.-50.0	0.-50.0	0.-40.6	VS	78.	1248.	2769.	0.0	10.0	3.0	-7.0
280	SI	4.-38.4	7.-20.3	0.-50.0	0.-50.0	0.-41.6	SI	1209.	2145.	2808.	0.0	-12.0	-9.0	-6.0
290	VS	27.-22.4	3.-27.3	0.-50.0	0.-50.0	0.-40.9	VS	1209.	2106.	3120.	0.0	-3.0	-8.0	-2.0
300	VT	15.-27.7	2.-31.1	1.-50.0	2.-43.5	2.-26.8	VT	1560.	2925.	3627.	0.0	-1.0	-3.0	0.0
310	VT	18.-25.7	2.-32.0	3.-36.6	5.-36.8	6.-15.4	VT	1131.	1677.	2574.	0.0	-12.0	0.0	-5.0
320	VT	16.-26.7	0.-19.1	4.-34.7	7.-33.8	6.-15.6	VT	780.	1599.	2535.	0.0	-12.0	1.0	5.0
330	V	123. -9.2	22.-16.2	62.-11.5	54.-15.8	12. -9.8	V	468.	1443.	2496.	0.0	9.0	21.0	11.0
340	V	179. -5.9	64. -8.8	160. -3.3	92.-11.1	15. -7.4	V	463.	1404.	2535.	0.0	12.0	34.0	13.0
350	V	166. -6.5	66. -8.5	114. -4.2	84.-12.9	18.-10.9	V	468.	1287.	2535.	0.0	14.0	13.0	12.0
360	V	97.-11.2	58. -5.8	62.-11.6	37.-19.3	4.-18.6	V	351.	1131.	2535.	0.0	12.0	14.0	9.0
370	VS	64.-14.8	42. -4.4	18.-22.1	6.-34.2	1.-32.8	VS	234.	1014.	2379.	0.0	7.0	12.0	-5.0
380	VS	54.-16.4	34. -6.3	13.-25.1	5.-36.7	0.-37.4	VS	234.	1014.	2028.	0.0	11.0	14.0	-6.0
390	VS	33.-23.6	30. -7.4	9.-28.2	4.-39.4	0.-39.2	VS	234.	1014.	2028.	0.0	2.0	-1.0	5.0
400	VS	27.-22.2	19.-11.5	4.-34.7	2.-43.5	0.-40.3	VS	273.	1053.	2028.	0.0	11.0	14.0	-9.0
410	VS	22.-24.1	27. -8.4	7.-32.9	2.-45.7	0.-39.4	VS	234.	1014.	2223.	0.0	-2.0	-7.0	-8.0
420	SI	5.-37.7	4.-24.1	1.-48.3	0.-50.0	0.-40.9	SI	234.	2028.	2457.	0.0	3.0	-8.0	-7.0
430	SI	2.-44.3	3.-28.2	0.-50.0	0.-50.0	0.-40.8	SI	390.	1443.	2223.	0.0	-11.0	-7.0	-6.0
440	UT	9.-32.0	1.-39.7	2.-43.2	4.-38.5	2.-24.2	UT	663.	1326.	1872.	0.0	15.0	9.0	11.0
450	UT	19.-25.6	4.-24.0	13.-25.2	11.-29.8	3.-22.8	UT	429.	1248.	2379.	0.0	7.0	13.0	0.0
460	VS	68.-14.4	24. -9.3	39.-15.5	17.-25.7	2.-26.9	VS	429.	1170.	2574.	0.0	21.0	13.0	22.0
470	V	135. -8.4	54. -8.3	85. -8.9	55.-13.5	9.-12.2	V	390.	1131.	2574.	0.0	15.0	17.0	19.0
480	V	116. -9.7	58. -8.9	84. -8.9	42.-17.6	5.-13.0	V	390.	1131.	2574.	0.0	15.0	14.0	12.0
490	V	95.-11.3	47. -8.4	79. -8.5	38.-19.8	9.-12.0	V	390.	1131.	2574.	0.0	11.0	13.0	5.0
500	V	92.-11.7	45. -8.8	69.-12.6	40.-19.4	8.-13.0	V	429.	1170.	2574.	0.0	13.0	23.0	4.0

Table 1 (Continued)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
TIME	CL	RAW	VOICE	LOW	MID	HIGH	CL	F1	F2	F3	RMS	A1	A2	A3
510	V	108. -6.4	41. -4.7	105. -7.8	97.-13.6	11.-18.2	V	585.	1248.	2535.	3.0	16.4	15.	14.
520	V	392. -1.4	48. -3.3	234. -2.8	261. -2.2	23. -3.5	V	663.	1287.	2535.	3.0	15.8	21.	14.
530	V	282. -2.3	54. -2.3	224. -2.4	281. -1.4	22. -4.0	V	702.	1326.	2574.	3.0	13.4	20.	15.
540	V	276. -2.1	47. -3.4	227. -1.1	264. -1.9	25. -2.9	V	741.	1365.	2574.	3.0	12.	17.	14.
550	V	291. -1.7	50. -3.0	197. -1.5	269. -1.8	27. -2.4	V	780.	1404.	2535.	3.0	13.8	21.	17.
560	V	353. 3.0	45. -3.8	230. -2.2	330. 0.2	35. 2.8	V	780.	1443.	2535.	3.0	17.8	25.	18.
570	V	386. -1.2	51. -2.4	207. -1.1	322. -2.8	32. -0.8	V	780.	1443.	2535.	3.0	14.2	24.	15.
580	V	274. -2.2	49. -3.0	184. -2.1	268. -1.8	28. -2.8	V	819.	1482.	2496.	3.0	12.8	18.	14.
590	V	255. -2.8	47. -3.5	161. -3.3	241. -2.7	25. -2.8	V	780.	1521.	2416.	3.0	17.	20.	14.
600	V	290. -1.7	41. -4.7	176. -2.5	278. -1.7	26. -2.6	V	741.	1521.	2223.	3.0	16.	25.	19.
610	V	238. -3.7	47. -3.4	158. -3.9	224. -3.4	24. -3.2	V	741.	1599.	2186.	3.2	13.7	24.	22.
620	V	189. -5.4	44. -4.3	115. -6.2	176. -5.4	24. -3.3	V	702.	1599.	2028.	3.0	12.8	21.	20.
630	V	136. -7.1	45. -3.9	82. -9.2	133. -7.9	23. -4.6	V	585.	1560.	1989.	3.0	16.8	16.	17.
640	V	96.-15.9	33. -6.4	38.-15.7	93.-16.0	9.-13.0	V	546.	1521.	2028.	3.0	9.8	6.	6.
650	VS	6.-35.2	7.-26.2	3.-37.5	3.-41.6	1.-35.4	VS	351.	1833.	2886.	3.0	3.8	3.	-7.
660	SI	5.-36.4	5.-23.2	1.-47.1	1.-50.0	0.-42.8	SI	1248.	1989.	2925.	3.0	-8.2	-10.	-5.
670	SI	7.-34.8	5.-23.7	3.-37.4	1.-50.0	0.-42.2	SI	312.	1053.	2067.	0.0	1.8	-8.	-12.
680	SI	2.-43.6	3.-26.1	1.-46.7	0.-50.0	0.-47.5	SI	1014.	1989.	2808.	0.2	-13.0	-13.	2.
690	SI	1.-48.6	1.-38.2	0.-50.0	0.-50.0	0.-39.4	SI	1911.	2457.	3471.	3.0	-7.0	-8.	-1.
700	SI	1.-50.0	1.-36.9	0.-50.0	0.-50.0	0.-38.9	SI	507.	2067.	3120.	0.0	-13.0	-5.	3.
710	SI	1.-50.0	1.-39.9	0.-50.0	0.-50.0	0.-42.1	SI	390.	1443.	2301.	0.0	-7.0	-11.	-9.
720	SI	1.-50.0	0.-47.1	0.-50.0	0.-50.0	0.-41.1	SI	1092.	2652.	3315.	0.0	-16.0	-4.	-5.
730	SI	1.-50.0	0.-45.3	0.-50.0	0.-50.0	0.-42.2	SI	1287.	2184.	3237.	0.2	-12.0	8.	24.
740	SI	1.-50.0	0.-46.4	0.-50.0	0.-50.0	0.-40.7	SI	1092.	1755.	2340.	0.0	-11.0	-6.	-2.
750	SI	1.-50.0	0.-44.5	0.-50.0	0.-50.0	0.-40.6	SI	624.	1482.	1911.	0.0	-8.0	-5.	-11.
760	T	7.-33.5	3.-43.7	3.-37.8	6.-35.2	1.-29.2	T	1209.	1833.	2613.	3.0	7.0	16.	6.
770	UT	12.-31.0	2.-38.5	6.-31.6	5.-36.3	2.-27.8	UT	273.	1248.	1872.	3.0	17.0	-4.	11.
780	V	77.-13.2	24. -9.3	58.-12.2	33.-19.9	1.-21.9	V	468.	1014.	2106.	3.0	16.0	14.	16.
790	V	96.-11.3	39. -5.1	82. -9.3	45.-17.2	3.-16.4	V	468.	975.	2496.	0.0	14.0	19.	8.
800	V	189.-10.2	44. -4.1	86. -8.7	49.-16.5	9.-11.8	V	468.	975.	2496.	3.0	18.0	37.	4.
810	V	144. -7.8	39. -5.1	98. -7.6	78.-13.5	11.-12.2	V	507.	1014.	2535.	3.0	22.2	19.	12.
820	V	148. -9.0	46. -3.7	132. -5.8	91.-12.2	10.-11.0	V	507.	975.	2535.	3.0	15.2	19.	9.
830	V	289. -4.5	45. -3.9	181. -2.2	175. -5.5	17. -6.2	V	624.	1092.	2535.	0.0	14.0	16.	4.
840	V	238. -3.4	48. -3.4	205. -1.2	219. -3.6	21. -4.4	V	663.	1170.	2535.	0.0	14.0	15.	9.
850	V	235. -3.5	47. -3.5	197. -1.5	214. -3.8	21. -4.3	V	663.	1170.	2535.	0.0	17.0	20.	15.
860	V	285. -1.9	42. -4.5	198. -1.5	226. -3.3	23. -3.7	V	702.	1170.	2535.	3.0	23.2	28.	17.
870	V	259. -2.7	42. -4.4	232. -0.1	272. -1.7	24. -3.3	V	741.	1170.	2574.	3.2	23.0	19.	12.
880	V	264. -2.5	45. -3.9	193. -1.7	247. -2.5	23. -4.8	V	741.	1170.	2574.	0.0	22.8	18.	11.
890	V	247. -3.1	41. -4.6	196. -1.6	238. -2.9	19. -5.3	V	741.	1170.	2535.	3.2	18.0	17.	12.
900	V	249. -3.0	43. -4.3	189. -1.9	235. -2.9	19. -5.2	V	780.	1209.	2535.	3.0	17.2	16.	9.
910	V	244. -3.2	43. -4.3	190. -1.8	232. -3.1	21. -4.4	V	780.	1209.	2496.	3.2	22.0	22.	14.
920	V	248. -3.1	39. -5.1	179. -2.3	218. -3.6	19. -5.1	V	780.	1248.	2496.	3.2	32.0	41.	14.
930	V	271. -2.3	38. -5.2	208. -1.2	265. -1.9	23. -1.5	V	780.	1287.	2457.	3.0	38.0	26.	31.
940	V	231. -3.7	43. -4.3	159. -3.4	215. -3.7	25. -2.6	V	741.	1287.	2418.	3.0	26.0	21.	24.
950	V	183. -5.7	43. -4.2	128. -5.3	169. -5.3	25. -2.0	V	741.	1326.	2379.	3.2	23.0	21.	16.
960	V	132. -8.7	42. -4.3	89. -7.4	129. -7.6	17. -6.4	V	741.	1365.	2340.	3.2	29.0	19.	14.
970	V	94.-11.5	44. -4.1	64.-11.3	69.-15.5	9.-12.8	V	702.	1443.	2145.	3.2	22.2	14.	4.
980	V	91.-10.8	33. -6.4	27.-18.9	21.-23.7	3.-21.2	V	468.	1443.	2223.	3.2	4.2	9.	14.
990	VS	7.-33.5	12.-14.6	3.-37.3	2.-44.6	1.-34.8	VS	507.	1365.	2418.	3.2	2.0	-5.	-4.
1000	SI	5.-37.7	4.-23.9	1.-52.9	0.-52.0	0.-42.3	SI	1443.	2340.	2925.	3.2	8.0	-4.	-8.
1010	SI	2.-46.4	2.-31.9	0.-58.0	0.-52.0	0.-39.5	SI	1755.	2418.	3354.	3.0	-3.0	-2.	-1.

Table 1 (Continued)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
TIME	CL	RAW	VOICE	LOW	MID	HIGH	CL	F1	F2	F3	RMS	A1	A2	A3
1020	SI	1.-49.4	1.-38.2	0.-50.0	0.-50.0	0.-40.0	SI	1443.	2145.	3120.	0.0	-4.0	-6.0	13.0
1030	SI	1.-50.0	1.-38.2	0.-50.0	0.-50.0	0.-40.3	SI	1521.	2457.	3120.	0.0	-5.0	0.2	0.0
1040	SI	1.-48.9	1.-40.7	0.-50.0	0.-50.0	0.-38.4	SI	1289.	1872.	2535.	0.0	-12.0	6.2	-5.0
1050	UT	10.-30.6	0.-44.7	2.-40.1	7.-32.9	2.-25.7	UT	782.	1521.	1872.	0.0	12.2	19.2	13.0
1060	UT	7.-33.7	2.-31.9	6.-32.3	7.-33.2	5.-17.8	UT	624.	1521.	2381.	0.0	-3.0	2.0	3.0
1070	SI	4.-36.2	2.-32.5	1.-49.1	1.-49.4	1.-33.3	SI	351.	1814.	1989.	0.2	6.2	-12.2	10.2
1080	VS	9.-31.9	6.-22.8	5.-32.6	2.-44.0	1.-34.7	VS	351.	1716.	2418.	0.0	-5.0	-11.2	-1.0
1090	SI	2.-44.9	2.-36.7	1.-48.4	0.-50.0	0.-40.4	SI	2184.	3042.	3588.	0.0	-3.0	-4.0	0.0
1100	SI	2.-44.0	1.-35.8	0.-50.0	0.-50.0	0.-42.4	SI	351.	1677.	2457.	0.0	-12.0	-9.0	-8.0
1110	SI	1.-47.9	1.-38.8	0.-50.0	0.-50.0	0.-40.6	SI	1365.	2262.	2808.	0.0	-5.0	-6.0	-7.0
1120	SI	1.-50.0	0.-48.4	0.-50.0	0.-50.0	0.-40.4	SI	1170.	2028.	2769.	0.0	-11.0	-4.0	-9.0
1130	SI	1.-50.0	0.-48.0	0.-50.0	0.-50.0	0.-41.0	SI	429.	2184.	2574.	0.0	-2.0	5.0	14.0
1140	VT	18.-26.1	6.-28.7	5.-32.6	6.-35.0	3.-28.3	VT	429.	2145.	2574.	0.0	17.0	5.0	5.0
1150	T	35.-20.1	21.-18.4	18.-22.2	12.-28.6	5.-17.2	T	390.	2106.	2574.	2.0	17.0	9.0	11.0
1160	VT	39.-19.0	15.-13.3	13.-25.4	0.-32.0	3.-22.0	VT	351.	819.	2067.	0.2	18.2	3.0	10.0
1170	V	66.-14.6	27. -8.4	31.-17.5	20.-24.5	6.-15.1	V	351.	858.	2145.	0.0	16.2	1.0	13.2
1180	V	88.-12.1	46. -3.7	49.-13.5	24.-22.6	7.-14.1	V	351.	936.	2028.	0.2	12.6	0.2	10.2
1190	V	102.-10.8	47. -3.5	66.-11.0	28.-21.5	9.-11.5	V	351.	975.	1989.	0.0	11.0	-2.0	8.0
1200	V	108.-10.3	57. -1.8	70.-10.5	28.-21.5	10.-10.6	V	390.	1131.	1911.	0.0	11.2	-2.0	9.2
1210	V	113. -9.9	57. -1.8	73.-10.1	29.-21.3	8.-12.8	V	390.	1794.	2730.	0.2	13.0	21.0	14.0
1220	V	117. -9.6	54. -2.2	66.-11.0	24.-22.7	5.-17.0	V	351.	1560.	2652.	0.0	17.0	28.0	22.0
1230	VS	109.-10.2	56. -1.9	46.-14.2	18.-25.4	2.-25.9	VS	312.	1677.	2457.	0.0	20.0	5.0	35.0
1240	VS	126. -9.0	68. -1.3	44.-14.6	18.-25.1	2.-26.2	VS	312.	1833.	2457.	0.0	21.2	2.0	24.0
1250	VS	119. -9.4	62. -1.1	46.-14.2	17.-25.8	2.-25.7	VS	312.	1794.	2496.	0.0	20.0	1.0	30.0
1260	VS	114. -9.8	67. -2.5	45.-14.3	17.-25.6	2.-26.9	VS	312.	1677.	2496.	0.2	18.0	1.0	18.0
1270	V	116. -9.7	65. -2.6	43.-14.6	16.-26.2	2.-24.5	V	312.	1560.	2496.	0.0	17.0	1.0	14.0
1280	V	110.-10.1	63. -0.9	41.-15.2	16.-26.4	2.-24.6	V	312.	1521.	2496.	0.0	17.0	6.0	13.0
1290	VS	109.-10.2	67. -0.4	42.-15.0	16.-26.4	2.-26.0	VS	312.	1443.	2496.	0.0	17.0	11.0	15.0
1300	VS	105.-10.6	65. -0.7	42.-14.9	17.-26.0	2.-25.8	VS	351.	1404.	2535.	0.0	16.0	4.0	12.0
1310	V	107.-10.3	53. -2.5	63.-11.5	31.-20.4	6.-15.7	V	390.	1443.	2535.	0.0	18.0	12.0	6.0
1320	V	131. -8.6	55. -2.2	86. -8.8	52.-16.0	9.-12.4	V	429.	1326.	2535.	0.0	20.0	19.0	14.0
1330	V	103.-10.7	56. -2.0	61.-11.7	39.-18.5	7.-14.3	V	351.	1248.	2535.	0.0	21.0	9.0	8.0
1340	VS	86.-12.3	56. -2.2	38.-17.9	11.-29.3	1.-29.2	VS	312.	1170.	2418.	0.0	17.0	6.0	9.0
1350	VS	72.-13.8	50. -2.8	26.-19.8	9.-31.2	1.-33.8	VS	273.	1131.	2223.	0.0	15.0	4.0	10.0
1360	VS	57.-15.8	43. -4.3	21.-21.8	7.-33.5	1.-35.8	VS	273.	1092.	2301.	0.0	11.0	4.0	9.0
1370	VS	42.-18.4	35. -6.1	15.-24.1	6.-35.2	0.-37.7	VS	273.	1092.	2301.	0.0	13.0	4.0	2.0
1380	VS	33.-20.6	28. -8.1	10.-27.4	4.-37.9	0.-38.4	VS	273.	1053.	2301.	0.0	10.0	3.0	23.0
1390	VS	29.-21.6	25. -8.8	8.-29.5	3.-40.1	0.-39.7	VS	273.	1014.	2145.	0.0	8.0	3.0	-7.0
1400	VS	26.-22.8	22.-10.1	7.-30.9	3.-41.7	0.-39.8	VS	273.	1014.	2067.	0.0	5.0	6.0	-7.0
1410	VS	20.-24.8	16.-12.9	6.-31.9	4.-39.3	1.-29.8	VS	507.	1053.	2418.	0.0	23.0	20.0	18.0
1420	V	111.-10.1	33. -6.5	95. -7.9	73.-13.1	6.-14.8	V	585.	1092.	2418.	0.0	17.0	15.0	12.0
1430	V	193. -5.2	48. -4.8	170. -2.8	159. -6.3	9.-12.1	V	624.	1131.	2379.	0.0	18.0	21.0	12.0
1440	V	202. -4.8	48. -3.3	168. -2.9	169. -5.8	10.-10.9	V	624.	1131.	2457.	0.0	23.0	24.0	15.0
1450	V	209. -4.6	47. -3.5	170. -2.8	175. -5.5	14. -7.9	V	663.	1170.	2457.	0.0	25.0	23.0	14.0
1460	V	218. -4.2	46. -3.6	171. -2.7	184. -5.1	13. -8.6	V	702.	1170.	2457.	0.0	26.0	26.0	23.0
1470	V	248. -3.1	47. -3.5	185. -2.1	204. -4.2	16. -6.8	V	741.	1170.	2496.	0.0	22.0	21.0	19.0
1480	V	241. -3.3	46. -3.6	193. -1.7	236. -2.9	21. -4.6	V	741.	1209.	2457.	0.0	20.0	18.0	24.0
1490	V	242. -3.4	46. -3.7	196. -1.6	228. -3.2	20. -4.7	V	741.	1229.	2457.	0.2	23.0	24.0	26.0
1500	V	224. -4.0	44. -4.1	178. -2.4	214. -3.8	17. -6.4	V	741.	1229.	2496.	0.2	22.0	22.0	19.0
1510	V	181. -5.9	42. -4.8	154. -3.6	173. -5.6	13. -8.8	V	780.	1289.	2496.	0.2	23.0	22.0	17.0
1520	V	134. -8.4	34. -6.3	133. -4.9	153. -6.7	20. -5.0	V	780.	1289.	2418.	0.0	20.0	19.0	9.0

Table 1 (Concluded)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
TIME	CL	RAW	VOICE	LOW	MID	HIGH	CL	F1	F2	F3	RMS	A1	A2	A3
1530	V	133. -8.5	28. -8.8	112. -8.5	129. -8.2	6.-14.8	V	780.	1240.	2457.	2.2	17.8	16.6	9.
1540	V	126. -9.0	24. -9.2	101. -7.3	120. -8.8	8.-12.4	V	780.	1240.	2418.	0.0	16.8	15.6	2.
1550	V	118.-10.1	23. -9.8	85. -8.9	108.-10.4	9.-11.9	V	819.	1240.	2418.	0.0	15.8	16.8	4.
1560	V	89.-12.8	15.-13.4	74.-10.1	87.-11.6	6.-15.6	V	780.	1240.	2418.	0.0	15.2	13.8	5.
1570	V	84.-12.5	14.-13.9	61.-11.6	77.-12.7	8.-13.2	V	819.	1287.	2418.	0.0	22.8	23.6	5.
1580	V	53.-16.5	10.-11.9	45.-14.4	56.-15.4	5.-17.3	V	858.	1287.	2418.	0.0	23.0	15.2	16.
1590	V	118. -9.6	12.-15.1	76. -9.8	110. -9.5	25. -2.9	V	819.	1287.	2340.	0.0	8.6	11.8	4.
1600	T	26.-22.8	12.-15.2	20.-21.5	27.-21.7	3.-20.8	T	819.	1326.	2301.	0.2	5.2	3.8	-4.
1610	VS	5.-37.8	6.-21.4	2.-41.1	2.-42.5	1.-36.3	VS	858.	1365.	2301.	0.0	4.4	2.8	-8.
1620	SI	3.-41.3	3.-26.3	0.-50.8	1.-50.8	0.-40.9	SI	741.	1404.	1872.	0.2	-2.6	19.4	2.
1630	UT	17.-26.4	4.-24.5	7.-38.4	15.-26.9	3.-21.1	UT	663.	1443.	2028.	0.0	2.2	3.2	22.
1640	VS	7.-34.1	3.-26.6	3.-38.2	2.-43.5	1.-35.4	VS	546.	1287.	2028.	0.0	1.8	-4.2	24.
1650	SI	4.-38.3	0.-21.3	1.-44.2	1.-50.8	0.-40.9	SI	390.	1443.	2184.	0.0	-2.8	-3.4	3.
1660	SI	2.-45.9	2.-29.1	0.-50.8	0.-50.8	0.-41.1	SI	117.	1287.	2613.	0.0	-7.8	-12.2	0.
1670	SI	1.-47.8	2.-33.2	0.-38.8	0.-50.8	0.-39.9	SI	1053.	2028.	3122.	0.0	-11.8	-7.6	2.
1680	SI	2.-46.2	1.-37.6	0.-50.8	0.-50.8	0.-39.2	SI	1209.	1989.	3081.	0.0	-1.6	-12.2	2.
1690	UT	3.-42.5	1.-35.7	0.-50.8	1.-49.6	2.-25.9	UT	1404.	3003.	3705.	0.0	15.6	-1.8	17.
1700	UT	12.-29.7	0.-44.8	4.-36.8	11.-29.7	2.-23.6	UT	663.	1248.	1521.	0.0	-1.8	15.8	16.
1710	T	9.-32.1	1.-36.2	6.-31.4	8.-32.6	1.-31.8	T	624.	1326.	1599.	0.0	3.8	20.8	32.
1720	UT	9.-32.3	1.-34.3	3.-38.9	7.-33.2	3.-21.4	UT	1404.	2808.	3159.	0.0	11.4	1.8	15.
1732	UT	12.-29.1	2.-31.7	2.-41.5	9.-32.8	5.-12.4	UT	858.	1521.	2184.	0.0	-22.8	-3.2	-4.
1740	UT	13.-28.8	1.-37.5	2.-41.2	7.-33.8	10.-11.8	UT	780.	1677.	2925.	0.0	-18.8	13.8	2.
1750	UT	15.-27.4	0.-44.9	2.-40.5	6.-34.1	11. -9.7	UT	858.	1716.	2574.	0.0	-14.8	-6.8	-1.
1760	UT	12.-29.6	1.-42.9	2.-39.7	6.-35.3	9.-11.8	UT	780.	1638.	2496.	0.0	-12.8	6.8	5.
1770	UT	13.-28.8	1.-41.6	1.-45.8	8.-37.1	10.-11.8	UT	1014.	1794.	3081.	0.0	-13.8	-7.6	3.
1780	UT	12.-29.2	0.-50.8	2.-40.5	5.-35.9	9.-11.4	UT	858.	1482.	2964.	0.0	-14.8	-14.2	16.
1790	UT	15.-27.3	0.-50.8	2.-41.8	5.-35.8	12. -9.3	UT	663.	1482.	2457.	0.0	-14.8	-3.2	3.
1800	UT	11.-29.8	1.-42.3	2.-40.4	5.-37.2	9.-11.7	UT	819.	1404.	2301.	0.2	-6.2	6.8	-4.
1810	UT	9.-31.5	1.-42.6	2.-43.6	4.-39.3	7.-13.7	UT	702.	1716.	2535.	0.0	-19.8	-9.4	3.
1820	UT	11.-30.8	0.-40.4	2.-42.6	4.-39.8	9.-12.6	UT	858.	1560.	2145.	0.0	-19.8	0.0	-12.
1830	UT	9.-32.4	0.-43.5	2.-40.9	3.-40.7	6.-15.1	UT	858.	1638.	2418.	0.0	-9.8	-4.2	-9.
1840	UT	9.-32.3	1.-40.9	2.-40.6	3.-39.9	7.-14.4	UT	663.	1521.	1989.	0.0	-12.8	-8.4	-14.
1850	UT	9.-31.9	1.-42.1	2.-41.8	4.-38.5	7.-14.3	UT	663.	1482.	2106.	0.2	2.0	-2.4	-9.
1860	UT	6.-34.7	1.-39.7	2.-41.8	3.-41.0	5.-17.6	UT	702.	1833.	2652.	0.0	-11.2	-1.8	14.
1870	UT	5.-36.5	0.-47.6	1.-44.6	2.-42.8	4.-18.5	UT	741.	1521.	2964.	0.0	-12.8	-4.8	-2.
1880	UT	6.-35.6	0.-47.3	2.-41.1	3.-40.9	5.-17.5	UT	780.	1404.	2496.	0.0	-10.8	0.0	14.
1890	UT	4.-39.4	0.-43.6	2.-41.9	2.-43.5	3.-22.6	UT	858.	1404.	2418.	0.0	-9.8	-6.2	-6.
1900	UT	3.-40.8	0.-43.8	1.-44.7	2.-45.7	2.-24.2	UT	624.	1521.	2457.	0.0	-9.8	3.8	8.
1910	T	3.-43.8	0.-43.8	2.-43.3	2.-46.0	1.-28.6	T	819.	1560.	2457.	0.0	3.8	17.8	-8.
1920	SI	2.-44.3	0.-44.5	1.-44.5	1.-47.6	1.-31.8	SI	624.	1404.	2067.	0.0	12.8	12.8	-7.
1930	SI	2.-43.2	1.-38.2	1.-47.3	1.-49.6	1.-34.8	SI	507.	1365.	2496.	0.2	-7.8	0.4	3.
1940	SI	1.-48.6	1.-35.8	1.-47.8	1.-50.8	1.-36.7	SI	1014.	1872.	3120.	0.0	-2.8	-7.8	9.
1950	SI	1.-47.5	1.-40.8	1.-48.2	1.-50.8	0.-37.2	SI	624.	1287.	1833.	0.2	-2.2	-5.8	2.
1960	SI	2.-44.4	1.-40.4	1.-49.8	1.-50.8	0.-38.2	SI	507.	1365.	1794.	0.2	-8.8	4.2	-4.
1970	SI	1.-50.8	1.-36.7	0.-50.8	1.-50.8	0.-39.9	SI	546.	1599.	2769.	0.2	-14.2	2.8	-1.
1980	SI	1.-49.8	0.-45.2	1.-52.8	1.-53.8	0.-39.5	SI	1053.	1794.	2730.	0.2	3.8	1.6	14.
1990	SI	1.-50.8	0.-48.6	1.-50.8	1.-51.8	0.-40.6	SI	585.	1521.	2496.	0.2	-7.2	8.0	1.
2000	SI	1.-53.8	1.-42.9	1.-50.8	1.-50.8	0.-41.1	SI	702.	1404.	2340.	0.2	-10.8	8.4	-9.
2010	SI	1.-49.6	0.-49.9	0.-52.8	2.-51.4	0.-39.9	SI	1209.	2418.	3393.	0.2	-10.8	3.8	11.
2020	SI	1.-48.2	0.-43.9	0.-52.8	3.-52.8	0.-39.7	SI	1716.	2535.	3393.	0.2	-8.2	-7.2	-7.
2030	SI	1.-49.8	0.-45.7	0.-52.8	0.-50.8	0.-40.9	SI	1053.	2106.	2808.	0.2	-3.2	-4.2	-7.

second value is the decibel equivalent of the first, normalized so that the largest decibel entry in each column is 0.0. Since all of the data in the table remain in disk storage, it is always possible to recover the normalization constant if necessary.

We do not contemplate adding any new basic capabilities for acoustic processing. Rather, we intend to develop algorithms that produce more precise acoustic parametrization using data provided by the existing procedures. We expect to refine our formant tracking and to establish a method for extracting pitch. However, since these and other changes are to be understood primarily in the context of the procedures for word verification, further elaboration is presented at the end of the next section.

E. Word Verification

Procedures for word verification relate the words predicted by syntactic and semantic processing to the acoustic data. The input to the word verifier from Pintle is a set of words that could be expected to occupy the next position in the utterance. The result of word verification is a subset, possibly empty, of the candidate words ordered according to a degree of agreement with the acoustic data at that location in the utterance.

Since Pintle is written in LISP and the acoustic processing is done in FORTRAN, it was necessary to develop procedures for communication between the two languages. An interface package, described in detail in Appendix B, makes it possible for a LISP program to create a fork (an independent process in the time-sharing system) containing a FORTRAN program, to share directly accessible data with that program, and to call functions in that program according to standard FORTRAN conventions.

For each candidate word, there is a function that tests for that particular word. The correspondence between the expected form specified

in the function and the contents of the acoustic stream is expressed as one of four confidence levels: positive, possible, unlikely, and impossible. For the first three levels, the function also returns an estimate of the ending position of the word in the acoustic stream.

The word verifier collects the results for each word in a set, eliminates the impossible words, and constructs a list ordering the rest of the words according to confidence level. The word with the highest ranking is returned to Pintle; any others are saved on a backup list to be used successively if their predecessor does not lead to the prediction of a new set of words, one or more of which can be found in the utterance. The ending position of the accepted word is used as the starting point for testing words in this new set.

To illustrate the word verification procedure, consider the sample sentence, PUT THE BLACK BLOCK IN THE BOX, in relation to the acoustic data contained in Table 1. In the analysis, BOX is one of the words predicted by Pintle at a location beginning approximately 1.34 seconds after the beginning of the utterance. This time is represented as 1340 milliseconds in Table 1. The word function for BOX produces the following actions.

1. It increments the time pointer by 170 milliseconds.
2. It attempts to find a vowel-like string in a 200-millisecond window centered at the incremented time pointer.
3. If Step 2 is successful:
   (a) It searches for a voiced stop ahead of the vowel-like string.
   (b) It searches for silence at the end of the vowel-like string. If a silence is found, it searches for unvoiced turbulence after the silence. It returns a confidence level, where appropriate, for each search.
4. It examines the vowel-like string as follows:
   (a) It calculates the average frequencies of the first and

second formants.

(b) It calculates the average slope of the first and second formants.

(c) It looks for discontinuities in the first and second formants.

If there are significant discontinuities or rapid changes in formant frequencies, it returns the value impossible.

5. It combines the results of the consonant search from Step 3 and the analysis of the vowel-like string in Step 4 as follows:

(a) If the average formant frequencies are reasonable for the vowel [ɑ] and all consonant searches are successful, it returns positive.

(b) If the average formant frequencies are reasonable, but a consonant search failed, it returns possible.

(c) If the average formant frequencies are unreasonable but all consonant searches are successful, it returns unlikely.

(d) If the average formant frequencies are unreasonable and at least one consonant search failed, it returns impossible.

In the example, the confidence level for BOX is positive. The results show a vowel-like string with first and second formant values consistent with [ɑ] in the interval 1420 to 1600, a voiced stop before the vowel-like string in the interval 1340 to 1410, silence after the vowel-like string from 1650 to 1690, and unvoiced turbulence from 1690 to 1910.

It should be clear that a word verification procedure of this kind was designed for use in a system with powerful syntactic and semantic constraints. In the analysis of the second noun group in the sample sentence, before BOX was confirmed, a set of adjectives was processed by the word verifier. All of the words were rejected except BLACK, for which the confidence level was unlikely. Pintle accepted BLACK tentatively, but that would have had to be the end of the sentence, and PUT THE BLACK BLOCK IN THE BLACK ... is syntactically and semantically unacceptable in the current system. Consequently, Pintle backtracked and looked for

nouns. Of the set predicted, BOX was confirmed with the highest confidence level. If BLOCKS had been a member of that set, the word verifier might have returned positive as well. However, since things can not be put in blocks, that word was excluded, on semantic grounds, from the set to be considered.

As indicated in the analysis of the sample sentence in Section II, the data used to test for the candidate words can be either acoustic or textual. If textual data are to be used, then the input is either a single typed word or a list of typed words. Each input word is analyzed to determine if it matches one of the candidate words from Pintle. Multiple matches make it possible to study Pintle's response to ambiguity in acoustic recognition without actually calling on the acoustic routines. The use of textual entries also makes it possible to study the word verification procedures for a variety of words in a particular context without requiring word functions to be written for all of the words that might be required in the processing of complete and meaningful sentences.

It is obvious that the word verification procedures in their present form would not allow subtle discriminations. However, the addition of more complex and powerful acoustic/phonetic rules to the analysis and decision-making parts of each word function should permit significant expansions of the system capabilities. The word verifier strategy is particularly appropriate for our system design, since Pintle operates on words rather than phonemes, allophones, or other phone-like units. Furthermore, the word verifier provides a way to deal with a significant subset of coarticulation problems that would be quite troublesome in a phoneme-verifier approach. For example, in testing for the sound [ℓ] in BLOCK, the effects of the preceding [b] and following [ɑ] can be incorporated into the word function. A generalized routine for verifying [ℓ] in random context is not required. Similarly, it is possible to use acoustic information about the end of the preceding word to influence

processing of the initial sounds in the current word. Coarticulation with following words is still a problem.

To a considerable extent, changes in the word verification procedures depend directly on increasing sophistication in acoustic processing. However, as indicated in the previous section, the need is not for new techniques for acoustic analysis but rather for ways to extract more information from the data we have. Furthermore, we believe that the motivation for changes should come primarily from the requirements of word verification. Our current efforts are directed toward providing many more subroutines for acoustic parametrization in order to refine the initial classification provided by the digital filter analysis and to provide additional formant data. For example, we are developing an algorithm to distinguish reliably among fricatives, specifically to separate [s], [ʃ], and [f-θ] (we do not expect to separate [f] and [θ]). We also are working on vowel segmentation and classification procedures that will extract boundaries within vowel-like strings, smooth formant curves, and plot slopes and standard deviations of formants. Our goal is to provide a variety of general procedures that can be used in the preparation of word functions for use in word verification.

F. Interactive Speech Analysis Facility

In the development of algorithms for acoustic parametrization and for word verification, we have made considerable use of the Interactive Man-Machine Speech Analysis System (IMMSAS). IMMSAS is a large FORTRAN program that handles data and process control for graphic display-oriented speech work. It operates on the PDP-10/PDP-15 computer facility with interaction through an Adage Graphics Display and a Teletype or similar terminal. IMMSAS contains a flexible, modular structure designed to accommodate present and future analysis techniques (e.g., digital filters, FFTs, and LPCs). Experiments can be

designed and run with real data; results are displayed graphically or on the line printer. A D/A converter provides an auditory response to verify the contents of a segment of digitized speech. A/D conversion, currently being done by a stand-alone program on the PDP-11, soon will be available through IMMSAS by microphone, thus facilitating data acquisition for exploratory testing.

IMMSAS has been used for the development of many acoustical processing algorithms. These algorithms, written as subroutines without local storage, can be run within IMMSAS or in a LISP/FORTRAN program structure. Furthermore, simple algorithms can be synthesized interactively from elementary array operations and FORTRAN functions. The interactive display facility has greatly facilitated debugging and program checkout.

The structure of IMMSAS is shown in Figure 4; single lines represent program control, and double lines show data flow. It will be useful to describe its operation in terms of data base management, graphic output, and interactive program control.

1. Data Base Management

There are two types of data handled by the system: time series data and processed data. Time series data represent the digitized output of A/D conversion with 9 to 12 significant bits. One utterance by a single speaker is assigned to one disk file. A record contains a description of the recording (speaker, sentence identifier, date of entry into the file system, sample rate, and number of samples). Two additional records can be used to indicate significant indices in a time series: one to point to events, such as phrase boundaries, word boundaries, or phonetic boundaries, and the other to mark pitch pulses.

Processed data result from submitting time series data to one of the processing modules. Each processed file is associated with its corresponding time series file. An identification record describes the

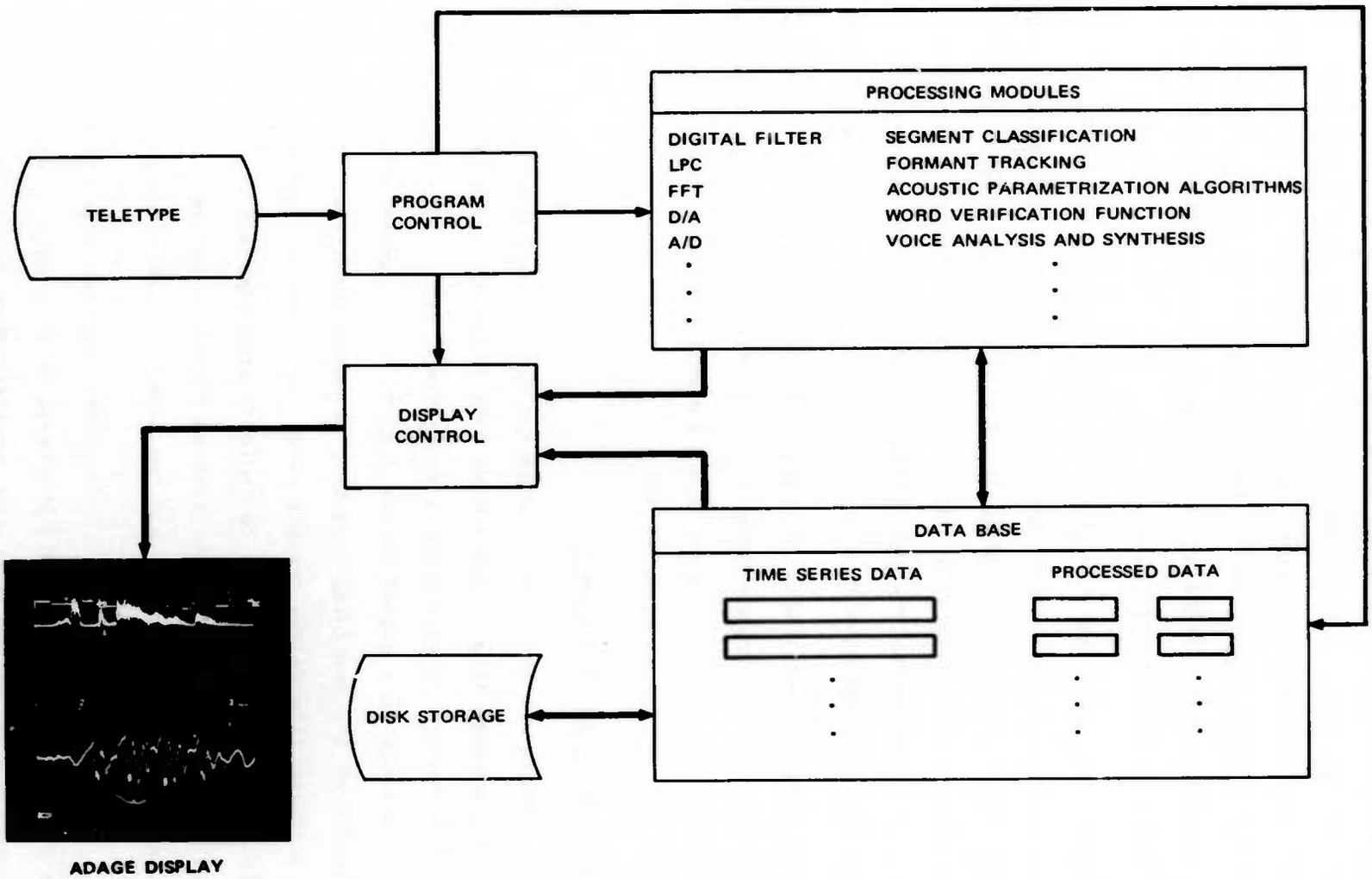

FIGURE 4 INTERACTIVE MAN-MACHINE SPEECH ANALYSIS SYSTEM ORGANIZATION

recording conditions and the contents of the rest of the file. The remainder of the file contains an arbitrary number of records of arbitrary length, each corresponding to a data array. For instance, the file contained in Table 1 resulted from the digital filter and LPC analyses of the time series data from the sample sentence discussed in Section II.

Both time series and processed data can be read from the disk into arrays organized to allow flexible referencing and dynamic allocation of available core.

2. Graphic Output

IMMSAS provides fast and flexible graphic output on the Adage display. For time series data, an rms envelope of the entire utterance can be shown on the upper half of the screen, while an expanded representation of a selected portion of the actual acoustical signal is shown on the lower half. Event markers and pitch markers can be displayed in a coordinated fashion on both the upper and lower traces.

Processed arrays are displayed using a general plotting routine that does automatic scaling and labeling. Data can be displayed in one dimension (in relation to an index) or as two-dimensional plots, e.g., a frequency spectrum or a distribution over time of the successive frequencies for the first three spectral peaks, as contained in Columns 9, 10, and 11 of Table 1. Control of various plot parameters, such as type of line (e.g., solid, dash, dot, blinking) and graph placement (e.g., coordination of rms envelope with formant frequency data, superimposition of two different analyses for the same raw data), provides flexibility in viewing the results of a processing step.

3. Interactive Program Control

Commands entered from the Teletype are used to control program flow: setting display and data parameters, specifying data input and

output, and calling for particular processing modules. The first part of each command is a mnemonic, followed by any required parameters, which are entered, free format, as floating point numbers or alphanumeric labels. Some parameters can be changed while a process is running by using the software pseudo-interrupt feature of the PDP-10.

A "mouse" connected to the Adage display can be used to interact with displays of time series data. Moving the mouse changes the location of a pointer on the face of the display. Pressing a button on top of the mouse assigns an index to a point in the time series; a line is entered on the display at that location, and a marker is entered in a corresponding location in the data file. Another button on the mouse allows scanning through the expanded representation of the time series data on the lower half of the display, either continuously, at periodic intervals, or discretely.

G. The SRI Question-Answering System

As indicated in the introduction to Section III, our initial intent was to pursue two separate system designs reflecting two different approaches to question answering. We chose to concentrate our resources on Pintle, accessing the data from acoustic processing through a word verification procedure. However, we did some preliminary work on modifications of an existing system so that it could be interfaced with a more traditional acoustic analysis procedure.

ENGSPK is the latest in a series of natural language, interpretive, question-answering systems being developed at SRI. Developed over a period of years by Coles and incorporating some early work by Raphael and Green (see Coles, 1972, for references), it integrates a formal theorem-prover with a procedure for natural language analysis to retrieve information from data files. In this system, the input sentence is analyzed by transformational and phrase structure procedures to identify

its syntactic structure. A semantic interpreter maps this structure into the first-order predicate calculus. A resolution theorem-prover is used to relate the resulting deep structure to axioms that represent the information stored in the data base. A generative component provides for a natural language response. Coles' most recent efforts have been directed toward determining the feasibility of using this system for querying a large data file (Coles, 1972).

The syntactic analysis component of ENGSPK is a bottom-up parser, dependent upon the availablility of an input string for processing. For use in a speech understanding system, ENGSPK would require a detailed analysis of the speech signal to generate hypotheses regarding the words in the message. These word hypotheses then would be checked against the accumulated syntactic and semantic data. In anticipation of the availability of such an acoustic processing component, some changes were made to allow for the accommodation of the system to speech input and to the "blocks world" problem domain. Three dictionaries were defined to satisfy the vocabulary requirements: the first contains 250 words, the second about 1000, the third is a domain-independent special list containing about 3000 verbs. In addition, a simple morpheme dictionary of prefixes and suffixes has been specified, and a scenario has been formulated for demonstration purposes. A technique for verb generalization using inflectional analysis now allows the system to represent in its memory verbs it has never encountered previously. The anaphoric reference subroutines have been improved to allow extended sequences of such references. Conversion of the whole system from Stanford-LISP to BBN-LISP also allowed substantial improvements in random access from secondary storage files.

Although we do not now intend to develop an appropriate acoustic processing component for ENGSPK, we do believe that work done by other ARPA contractors may prove relevant for this purpose. To the extent

that a suitable word hypothesizer becomes available, the development of a system linking it with ENGSPK would be a worthwhile effort. We are willing to cooperate with other contractors toward accomplishment of this goal.

IV DIRECTIONS FOR CONTINUING RESEARCH AND DEVELOPMENT

A. Overview

Our program for continuing research and development centers on the implementation of an integrated speech understanding system. We do not expect to make any substantial change in the system design until we have thoroughly investigated our current strategy. As indicated in the previous sections, we recognize the need for further work on Pintle, on algorithms for acoustic parametrization, and on word verification procedures. Most important will be the effects on the system from an increasing interdependence of these components as we experiment, make modifications, and experiment again. Additional work on semantics and pragmatics, on prosodics, on a grammar for spoken English, and on techniques for speaker independence can be expected to have a significant impact on increasing the sophistication of the system. We also are exploring ways of increasing the richness of the problem domain so that it can sustain more complex goal-oriented interactions with a user.

B. An Integrated Speech Understanding System

The communication between the programs for syntactic and semantic analysis and the acoustic data through the procedures for word verification is relatively simple at present. Pintle predicts a set of words, the word verifier tests the words against the acoustic data, a word is selected according to degree of agreement, and Pintle continues accordingly. It is reasonable to expect that communication should be more complex. Prosodic information can affect high-level decisions in Pintle, and feedback of acoustic data within the word verification procedure can result in more economical searches. Consequently, although we are

now independently making changes in Pintle and in the word verifier, it is likely that future modification will reflect the increasing interdependence of procedures within the system.

Following the analysis of the sample sentence in Section II, some provisional features of the current implementation were noted. We do expect to be completely "on-line" relatively soon, i.e., to have the A/D conversion take place wholly within our PDP-10 computer rather than having to transfer digitized tapes from the PDP-11. When the A/D programs are available, we will be able to enter voice directly into the system. However, this capability will not make a major difference in exercising the system itself in the near future because of the time required for A/D conversion and for other basic acoustic data processing. Still, it will simplify debugging algorithms to be used in word verification. The availablility of a quiet room for recording in close proximity to our display facility and to the computer should increase our efficiency, but it also will have little actual effect on the system itself for the present.

1. Pintle

To achieve greater flexibility in our procedures for syntactic and semantic analysis, we are beginning to test some new algorithms in QA4. They must be able to use information from word verification, from semantics, and from prosodic analysis to control the path through the grammar. We want to be able to assign priorities to the alternatives at any particular choice point. In effect, we are trying to find the minimal cost path through the graph that represents our grammar.

2. Acoustic Processing

As indicated in the previous section, no new basic capabilities for acoustic processing will be developed. Rather, we will be trying to extract more useful information from the data now available to us.

In relation to the procedures for word verification, we need more precise techniques for parametrization to allow us to distinguish more clearly among the words in a set predicted by Pintle. For prosodic analysis--i.e., to identify stressed words, intonation contours, and linguistically significant pauses--it will be necessary to process acoustic data over substantial periods of an utterance. In this work, we expect to cooperate closely with the University of Michigan and with UNIVAC.

3. Word Verification

With the availability of algorithms for making more precise classifications of sounds, it should be easier to write more word functions. Having written 100 or so, we should be able to establish some general procedures for creating them. More important, we will have used enough to be able to judge how well the word verifier concept actually works. We believe that word functions provide a particularly good medium for embodying acoustic phonetic rules. Coarticulation effects within words can be handled easily, and we are interested in determining how well the procedure works for coarticulation between words. Feedback of acoustic data from word verification is another major problem. It does seem likely that such information can help in testing successive words within a set, but we are interested in determining whether it can be used more generally to influence Pintle.

C. Semantics and Pragmatics

The development of effective procedures for handling semantics and pragmatics is essential both for the operation of an initial speech understanding system and for extrapolating the results of our efforts to other problem domains. Recent research in syntax has provided some very general and reasonably efficient procedures for parsing that (in principle) can handle perhaps all of the syntactic constructions in

English. However, it is not clear that existing parsers can accommodate the semantic information that we believe is necessary for our system. Our continuing work on Pintle is intended to provide flexibility and control over priorities assigned to choice points in the grammar. Now we need procedures that model the world and the user to provide the proper kind of guidance for assigning those priorities.

The point of departure for our work on semantics and pragmatics is the model of case grammars being developed by Charles Fillmore (1971a, 1971b) of the University of California at Berkeley. This model is being extended by him and others toward "a fully developed system of linguistic description" involving the analysis of complex utterances or messages, not just sentences, and including multiple participants in conversations. Fillmore's work builds on transformational grammar but extends beyond competence into performance in a social context.

Even without all of the possible elaborations, Fillmore's case concept has a direct implication for the kind of semantic structures that we intend to incorporate into revisions of Pintle. In case grammar, the propositional core of a simple sentence is a predicator (verb, noun, or adjective) in construction with one or more entities, each related to the predicator in one of a set of semantic functions known as cases. The cases identify particular roles, such as instigator of an action, experiencer of an event, object undergoing change or movement, and location or time of an event. The number of cases proposed is small, and there are some markers in the surface structure (e.g., prepositions) that make the application of these linguistic concepts to recognition a realistic enterprise.

D. Prosodic Analysis and a Grammar for Spoken English

The importance of prosodic information already has been mentioned several times in this section. It is particularly important for the SRI

speech understanding system because it can provide a basis for assigning priorities in Pintle at the beginning of an utterance. It also can suggest hypotheses about the presence of syntactic or semantic "boundaries" in an utterance that can be used in determining the most likely path through the grammar. More generally, however, the rules for such speech elements as stress, intonation, and pauses should form part of a grammar for spoken English. This grammar also should incorporate phonological and acoustic/phonetic rules that describe how people actually speak.

We can expect that the artificiality both of the problem domains chosen for speech understanding research and of the interaction with a computer through a microphone will limit the range of English spoken and the variety of styles used. However, it is obvious that we need to maintain close communication with linguistics because even our more limited tasks can be accomplished only through cooperative efforts. Coordination within the ARPA Speech Understanding Research Program needs to be complemented by extensive contacts with other linguists.

E. Speaker Independence

In their present form, our feature-detection algorithms have not been talker-dependent, principally because they have been designed to make reliable--if crude--classifications. As the algorithms become more complex, it is likely that variations among speakers will create perturbations in the results from the word functions. The first attempts to gain acceptable speaker-independent performance will probably involve manual threshold adjustments of the variables defined in the word verification procedures. As familiarity with the data base and with the procedures themselves increases, attempts will be made to introduce some automatic "speaker normalization" calculations.

Appendix A

DIGITAL FILTER DESIGN PROGRAM

Technology Service Corporation of Santa Monica, California, prepared a digital filter design program for SRI that will design and verify recursive digital filters. The program is user oriented to permit inexperienced programmers and designers to use it. Included in the program are three standard filter prototypes: Butterworth, Bessel, and elliptic (Cauer parameter) designs. Other filter functions may be supplied externally by a specification of the poles and zeroes of the transfer function.

The program has the capability of transforming any of the above standard designs to bandpass, bandstop, low-pass, and high-pass transfer functions with arbitrary cutoff frequencies. In addition, the program determines recursive sampled-data or digital representations for any of these filter transfer functions using either the standard z-transform or the bilinear z-transform. Program output includes poles and zeroes of the above continuous functions, the coefficients for the recursive digital filter functions realized in parallel form, and optionally printed frequency and time response characteristics, as well as a FORTRAN function representation of the digital filter. The frequency and time response characteristics may be displayed by the graphics system associated with the computer.

Appendix B

A LISP-FORTRAN INTERFACE

The interface facility provides for:

- Creation from LISP of subfork(s) containing FORTRAN programs.
- Subprogram calls from LISP to the FORTRAN fork.
- Creation of REAL or INTEGER arrays accessible from both forks.

1. The FORTRAN Side

There are only a few requirements for the FORTRAN program:

- The symbol table produced by the loader must be saved with the program.
- A small MACRO interface package must be loaded with the program.
- Several entry vector locations (Numbers 3 to 6) must be left free for use in interfork transfers. (Programs normally use Locations 0 and 1 only).
- If arrays are to be shared with LISP, then a block of pages to hold the arrays must be specified when the program is saved.

The steps to create the FORTRAN program are as follows:

- Load the program with symbols (i.e., use /B/S) and include ⟨RIDDER⟩F4ØF.REL. When the loader exits, go into DDT, do any pre-save initialization necessary, and then transfer to the interface package by typing FKINIT$G. It will respond with NUMBER OF PAGES SHARED: and wait for you to type in a number (in OCTAL) indicating how many pages will be shared with LISP.
- If this count is non-zero, then the program will type FIRST SHARED PAGE: and wait for another number (also in OCTAL) indicating the first page in the FORTRAN address space of the block of pages to be shared.
- Control then returns to the EXEC. Save the program and symbols using either SAVE or SSAVE. (DDT need not be saved--it will be

automatically brought in if needed at run time).

2. The LISP Side

The following functions constitute the LISP side of the interface. (Load ⟨RIDDER⟩FORK.COM to use them).

fkinit [program]

This function is an nlambda that creates and initializes a fork containing PROGRAM. Information about the fork is saved on FORKDATA. (FORKDATA is used as a free variable by the other functions to access the data associated with the fork. This makes it possible to talk about multiple forks by changing the binding of FORKDATA).

Example: (FKINIT ⟨RIDDER⟩IMMSAS.SAV)

fkddt[]

This function transfers control to the fork DDT and waits for the fork to halt. It is not necessary to have saved DDT with the program. To return to LISP from the fork, type HALT$G to DDT.

Example:

←	FKDDT()	From LISP go to DDT in fork
	X+1$1B HALT$G	Set a breakpoint, then return to LISP
	T	Value of FKDDT is T
←	FKCALL (XFCN REAL)	From LISP call the fork subroutine (FKCALL is discussed below)
	$1B⟩⟩X+1 ... $P	Stop at breakpoint then proceed
	3.1415979	FKCALL types result
←		Back in LISP

fkkill[]

This function kills the fork and sets FORKDATA to NIL.

fkarray [id;type;size]

This function is an nlambda that creates a shared array. SIZE is evaluated and a block of SIZE+1 words is allocated in the pages shared with the fork. ID is set to the LISP address of the array and entered into a LISP hash table of symbols for the fork with value equal to the fork address of the first data word of the array. TYPE can be either REAL or INTEGER and specifies the type of number to be stored. The value of FKARRAY is the LISP address of the array.

NOTE: shared arrays are not garbage collected. They stay around until the fork is killed.

Example: (FKARRAY SHR REAL 100)

fkarraysize[a]

This function returns T iff A is a shared array for the fork specified by FORKNAME.

fkelt[a;n]

This function returns the Nth element of shared array A. Its value is a boxed integer or a boxed floating point number according to the type of A.

fkseta[a;n;v]

This function stores V into the Nth element of shared array A. Its value is V.

fksym[id]

This function asserts that ID is either a global symbol in the fork or the name of a shared array. Its value is the fork address of ID (as a boxed integer). FKSYM first looks for ID in the hash table of symbols for the fork. If ID is not found there, then the symbol table saved with the fork is searched for a global definition of ID, and the resulting value is entered in the hash table. If ID cannot be found, then FKSYM calls error.

Note that to reference an ID declared in the fork, it must be a global symbol. This means it must be in named COMMON with

ID the name of the COMMON block (i.e., declared as COMMON /ID/ID). This is the same convention that must be followed if ID is to be referenced from a MACRO program loaded with the FORTRAN program.

Example: (FKSYM (QUOTE F4ARRY))

fkcall[id; type;argl ... argn]

This function is a nospread nlambda (i.e., it can have an arbitrary number of arguments, and it gets them in an unevaluated form) that calls a FORTRAN subprogram and returns the result.

The value of ID is used as the name of the subprogram. The function FKSYM is called to find the fork address.

The value of TYPE specifies the result type of the subprogram. The result types currently accepted are:

INTEGER--value is boxed integer
REAL--value is boxed floating point number
LOGICAL--value is T or NIL
SUBR--value is NIL

The following process is carried out to determine the meaning of each subprogram argument.

If ARGi is of the form (BIAS A N), then A is evaluated and converted to a fork address by FKSYM. The sum of (eval N) -1 and the fork address for A is used to specify the parameter. This mirrors the use of A[N] as an argument in FORTRAN.

If ARGi is a list of the form (INTEGER X), (REAL X), or (LOGICAL X), where X is any atom, then the value of X is passed, and later X is set to the value assigned by the subprogram to its corresponding dummy argument. This makes it possible to return multiple results from subprograms in the standard FORTRAN manner.

In case ARGi is not one of the above, it is evaluated, and the type of the value is used to determine the manner in which the argument is passed.

If the value is a logical quantity (T or NIL) or a number, the appropriate value is used (-1 for T, 0 for NIL, the unboxed value for a number).

If the value is a LISP array, then a copy of the entire array is passed. (Needless to say, it should be a small array.)

If the value is a string, then a copy of the string (left-justified, 7-bit ASCII, and followed by a word containing 0) is passed.

If the value is the address of a shared array, then the fork address is used.

If the value is an atom the FKSYM is called and its value is used to specify the parameter.

Otherwise FKCALL calls error.

NOTE: lists cannot be sent--use shared arrays to create "lists" for FORTRAN.

Rough timings indicate that the simplest FKCALL, namely a call to a SUBR of no arguments, takes about 15 milliseconds, of which about 10 are spent executing the appropriate monitor JSYSs. Each argument for the subprogram adds about 2 milliseconds, as does returning a result.

The argument information must fit in 14 words, since it is sent to the lower fork in the accumulators (the other two accumulators are used for specifying the argument types and the subprogram address). LISP array arguments take arraysize words; strings take (nchar+4)/5 + 1 words; all other argument types take a single word.

Example:

```
(FKARRAY IA-LISP INTEGER 100)
(FKSYM (QUOTE IA))
(FKCALL (QUOTE MOVARY) (QUOTE SUBR) (BIAS (QUOTE IA) 100) IA-LISP 100)
(FKCALL)  (QUOTE SUM)  (QUOTE INTEGER) IA-LISP 10 50).
(FKCALL (QUOTE AVG) (QUOTE REAL) (QUOTE IA) 100 (REAL DEVIATION))
```

NOTE: a minor annoyance occurs when FORTRAN does teletype I/O because the job global TTY mode information is changed in a way that interferes with standard LISP interaction. The primary symptom of this is the failure of LISP to respond when the closing right parenthesis is input. To restore the proper settings, simply type a control-E to LISP (part of the processing of control-E resets the TTY mode to its original value).

Alternatively, the problem can be eliminated by use of the LISP (nlambda) function FKX, which saves the TTY mode, evaluates its argument, and restores the mode.

Example:

```
(FKX (FKCALL (QUOTE PRIN) (QUOTE SUBR) "('ØHELLO')"))
```

PRESENTATIONS

1. D. E. Walker, "Human Speech: In Recognition of the Problems Involved in Its Understanding," IEEE Systems, Man, and Cybernetics Chapter, Menlo Park, California, 8 May 1972.

2. (same), Queens College of the City University of New York, Flushing New York, 16 May 1972.

3. (same), University of California, Berkeley, California, 31 May 1972.

4. D. E. Walker, "Speech Understanding and Computational Linguistics," Annual Meeting of the Association for Computational Linguistics, Chapel Hill, North Carolina, 27 July 1972.

REFERENCES

1. Coles, L. S., "Techniques for Information Retrieval Using an Inferential Question-Answering System with Natural-Language Input," Technical Note 74, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (November 1972).

2. Fillmore, C. J., "On a Fully Developed System of Linguistic Description," in "Feasibility Study on Fully Automatic High Quality Translation," W. P. Lehmann and R. Stachowitz, eds., Vol. I, pp. 77-94, RADC-TR-71-295, Griffiss Air Force Base, Rome Air Development Center (December 1971 (a)).

3. _____, "Some Problems for Case Grammar," in 22nd Annual Round Table on Languages and Linguistics, R. J. O'Brien, ed., pp. 35-56 (Georgetown University Press, Washington, D.C.,) (1971 (b)).

4. Hudson, R. A., English Complex Sentences (North-Holland Publishing Company, Amsterdam, Netherlands, 1971).

5. Markel, J. D., "Formant Trajectory Estimation from a Linear Least-Squares Inverse Filter Formulation," SCRL Monograph No. 7, Speech Communication Research Laboratory, Santa Barbara, California (October 1971).

6. Newell, A., et al., "Speech Understanding Systems: Final Report of a Study Group," Carnegie-Mellon University, Pittsburgh, Pennsylvania (May 1971). To be published by North-Holland Publishing Company, Amsterdam, Netherlands, 1973.

7. Rulifson, J. F., Derksen, J. A., and Waldinger, R. J., "QA4: A Procedural Calculus for Intuitive Reasoning," Technical Note 73, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (November 1972).

8. Winograd, T., "Procedures as Representation for Data in a Computer Program for Understanding Natural Language," Report MAC-TR-84, MIT Project MAC, Massachusetts Institute of Technology, Cambridge, Massachusetts (February 1971). Published as Understanding Natural Language (Academic Press, New York, New York, 1972).